



КУРСОВ ПРОЕКТ

Онтология на криминална случка

Факултет по Математика и Информатика

Студент: Борислав Стоянов Марков

Факултетен номер: 0MI3400048

Учебен план: Изкуствен Интелект (редовно, магистър)

Курс: Курс 1; **Група:** Група 1

Активен период: 2021/2022 зимен, магистри

Дисциплина: Представяне и Моделиране на Знания

Дата: 30.12.2021г.



1. Съдържание

1. Съдържание	2
2. Увод.....	2
3. Сапунена криминална случка	3
4. Онтология на затворен свят	4
4.1 Елементи на началната онтология	4
4.2. Вторична онтология на базата на правила	8
4.3 Изводи със SPARQL ^[4]	10
5. Реализация на проекта.....	11
6. Идеи за бъдещо развитие и подобрения	12
7. Източници и използвана литература.....	13
Приложения.....	13
1. Сурс код (Source code).....	13

2. Увод

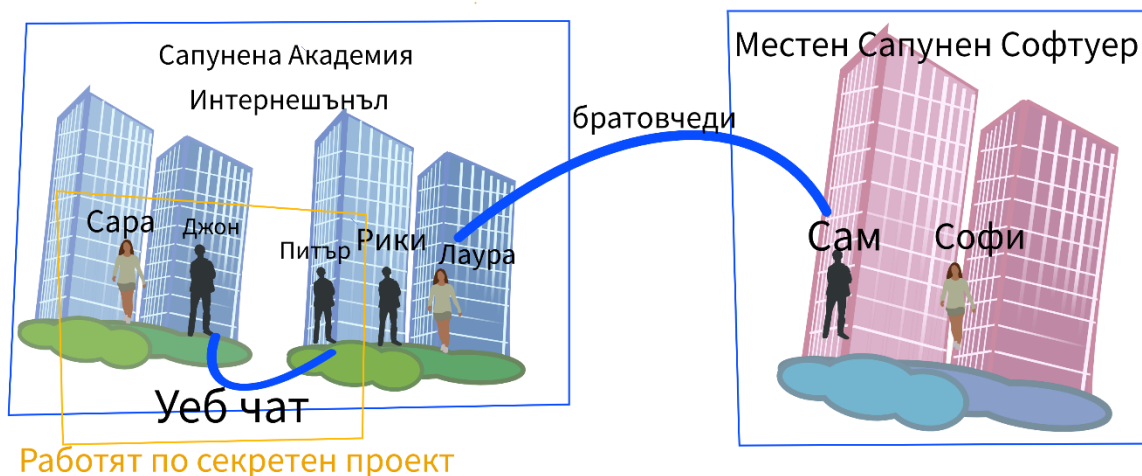
Идеята за проекта е взета от сапунената опера от книгата „Knowledge Representation and Reasoning“ (Brachman и Levesque)^[1]. В нашият случай – в града се е случила криминална случка. Има изтичане на информация от една софтуерна фирма и градският детектив разпитва заподозрените. Предстои да опишем това с две онтологии. Ще проследим кой кого познава и така ще помогнем на детектива да стигне до заподозрените. Описателният език, който ще използваме е OWL DL^[3], използваната програмна библиотека е Apache Jena. Ще използваме наши правила, описани във файл, за да направим изводи (Reasoning). Имаме начална и допълнителна онтология. Началната онтология поставя класовете, свойствата и индивидите, а в последствие на базата на правилата донасяваме още свойства, улесняващи крайния извод. Накрая ще направим няколко извода със SPARQL от локален контекст, без сървър.

Целите на проекта са да се покаже как може да се използват знанията и да се правят изводи, когато това не е очевидно. Областта може да е криминалистика, но не се ограничаваме само до нея.

3. Сапунена криминална случка

Джон(John) работи във софтуерната фирма „Сапунена Академия Интернешънъл“ (Soap Academy International). Проектът на Джон е конфиденциален и се казва „Робо-Сапунена Стратегия“ (Robo Soap Strategy). Джон има колега, с когото добре се познават, който се казва Питър (Peter) и работят по един проект. Пишат си постоянно във фирмения чат. Джон и Питър имат колега Рики (Riki), който се познава с Питър, но не се познават с Джон лично. Джон има също и колежка Сара(Sara), която работи по същият проект като него. Рики не работи по проекта „Робо-Сапунена Стратегия“. Лаура тайно харесва Джон, но е в друг офис, не работят заедно и се виждат само по фирмени събирания или на онлайн срещи. Джон не харесва Лаура и не откликва на чувствата ѝ. От друга страна Рики харесва Лаура, която не му обръща внимание. Рики е готов на всичко да я заинтересува. От друга фирма „Местен Сапунен Софтуер“ (Domestic Soap Software), конкурент на Сапунена Академия Интернешънъл, са получили информация за проекта Робо-Сапунена Стратегия и информацията е обявена публично. Местният детектив разпитва участниците за случките през последната седмица и установява, че Джон не работи в една сграда с Питър. Питър работи в една сграда с Лаура и Рики. Лаура харесва Джон. Питър харесва Лаура, но Лаура него – не . Рики работи зад гърба на Питър и може да види екрана му. Питър не заключва монитора си, когато ходи до кухнята в офиса. Вечерта преди да изтече информацията очевидци са видели Лаура да говори с братовчед си Сам (Sam). Сам работи за конкурентната фирма „Местен Сапунен Софтуер“. Софи(Sophie) и Сам са колеги. Лаура и Рики постоянно си говорят в офиса в кухнята.

На Рисунка 3.1 е онагледена случката. Изобразени са само базовите факти, а допълнителните ще бъдат изложени по-надолу поетапно.



Рисунка 3.1

4. Онтология на затворен свят

За да моделираме случката можем да използваме OWL DL, тъй като „OWL Lite“ не предоставя достатъчно функции за задачата. Ще представим таблично класовете и атрибутите към тях, описващи само това, което ние е необходимо за да намерим теча на информация. В началото ние създаваме една базова онтология, която ще наричаме **начална онтология**. След това добавяме със средствата на Apache Jena допълнителни знания под формата на правила и получаваме **вторична онтология**, допълнена с атрибути и генерирани класове, подпомагащи изводите.

4.1 Елементи на началната онтология

В таблица 4.1.1 са дадени класовете на началната онтология, според описаната история в т.3.

$\text{Person} \sqsubseteq \text{Thing}$
$\text{Man} \sqsubseteq \text{Person}$
$\text{Woman} \sqsubseteq \text{Person}, \text{complementOf}(\text{Man})$
$\text{Company} \sqsubseteq \text{Thing}$
$\text{Project} \sqsubseteq \text{Thing}$
$\text{Event} \sqsubseteq \text{Thing}$



WitnessedEvent \sqsubseteq Event
Conversation \sqsubseteq WitnessedEvent
OnlineConversation \sqsubseteq Conversation

Таблица 4.1.1

В таблица 4.1.2 са дадени свойствата на класовете на началната онтология. Свойството **canShareSensitiveInfoIndirectTo** ще бъде използвано в следващата онтология и ще бъде сложено автоматично с програмен код на базата на правила.

Domain	Property	Range	Characteristics
Person	worksFor	Company, Project	
Conversation	hasConversationParticipant	Person	<i>inverseOf</i> <i>isPartOfConversation</i>
Person	isPartOfConversation	Conversation	<i>inverseOf</i> <i>hasConversationParticipant</i>
Person	canSeeMonitor	Person	-
Person	isGoodFriendTo	Person	<i>symmetric</i>
Person	canShareSensitiveInfoTo	Person	
Person	canShareSensitiveInfoIndirectTo	Person	
Person	loveButNotLoved	Person	<i>functional</i>

Таблица 4.1.2

В Таблица 4.1.3 са дадени индивидите използвани в случката и техните свойства. Ограничили сме описанието само до ниво, за което можем да направим съответния извод. Например това, кои хора работят в една сграда, не е необходимо да се описва.

john \rightarrow [AND Man [FILLS :worksFor soapAcademyIntCompany] [FILLS :worksFor roboSoapStrategyProject] [FILLS : isPartOfConversation peterAndJohnChat]]
sara \rightarrow [AND Woman [FILLS :worksFor soapAcademyIntCompany] [FILLS :worksFor roboSoapStrategyProject]]



peter → [AND Man [FILLS :worksFor soapAcademyIntCompany] [FILLS :worksFor roboSoapStrategyProject] [FILLS : loveButNotLoved laura] [FILLS : isPartOfConversation peterAndJohnChat]]
riki → [AND Man [FILLS :worksFor soapAcademyIntCompany] [FILLS : canSeeMonitor peter] [FILLS : isGoodFriendTo laura] [FILLS : isPartOfConversation kitchenConversation]]
laura → [AND Woman [FILLS :worksFor soapAcademyIntCompany] [FILLS :loveButNotLoved john] [FILLS :isGoodFriendTo sam] [FILLS :isGoodFriendTo riki] [FILLS : isPartOfConversation eveningConversation] [FILLS : isPartOfConversation kitchenConversation]]
sam → [AND Man [FILLS :worksFor domesticSoapSoftwareCompany] [FILLS : isPartOfConversation eveningConversation] [FILLS : isGoodFriendTo laura]]
sophie → [AND Woman [FILLS :worksFor domesticSoapSoftwareCompany]]
soapAcademyIntCompany → Company
domesticSoapSoftwareCompany → Company
roboSoapStrategyProject → Project
eveningConversation → [AND Conversation [FILLS : hasConversationParticipant laura] [FILLS : hasConversationParticipant sam]]
kitchenConversation → [AND Conversation [FILLS : hasConversationParticipant laura] [FILLS : hasConversationParticipant riki]]

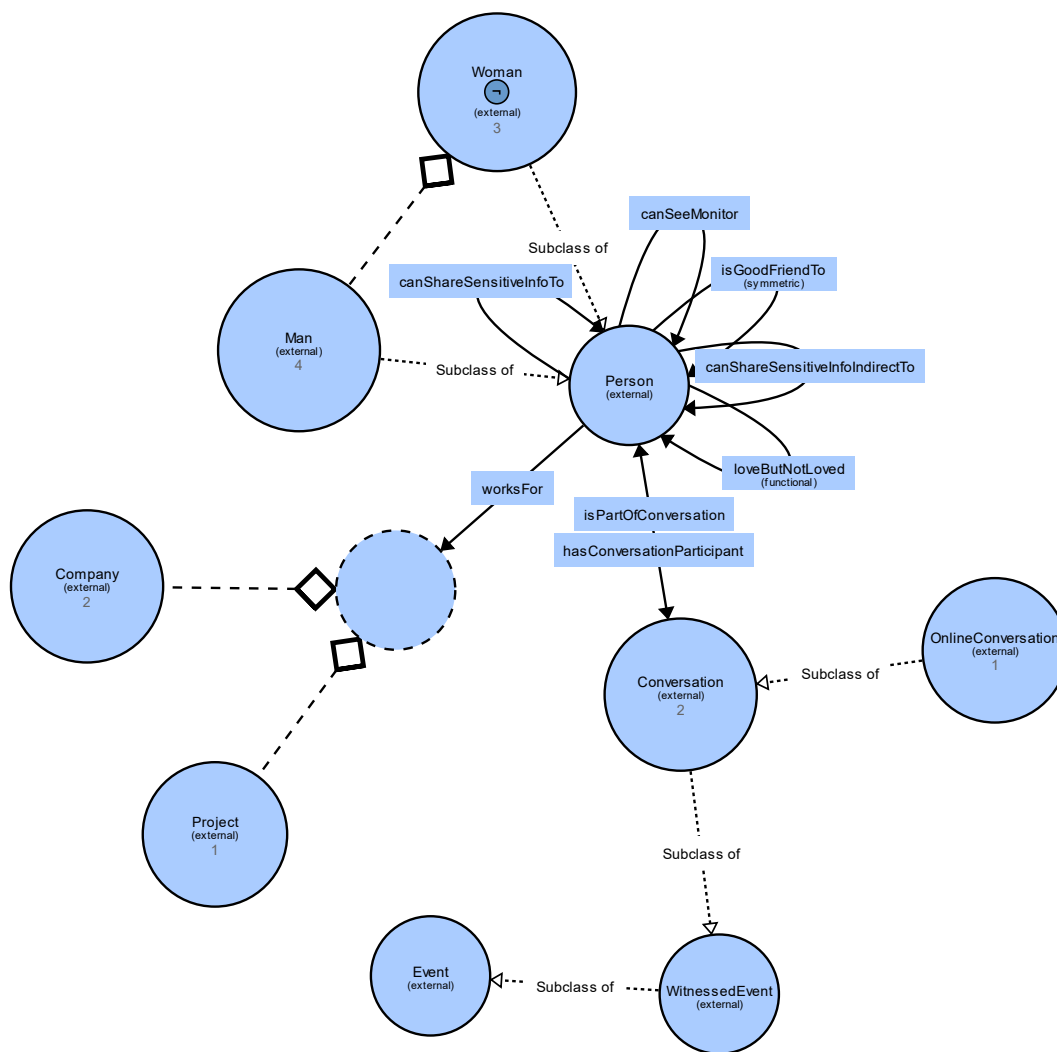
peterAndJohnChat → [AND OnlineConversation

[FILLS : hasConversationParticipant peter]

[FILLS : hasConversationParticipant john]]

Таблица 4.1.3

Ще представим и визуализация на онтологията на фигура 4.1.4.



Фигура 4.1.4

За да се направи качествен извод, от къде би могло да изтече информация, са необходими определени улеснения. Ще използваме една функция на Apache Jena, която може да наслага определени атрибути на базата на зададени правила (Reasoning).

4.2. Вторична онтология на базата на правила

Като допълнителен извод искаме да намерим всички които биха могли да споделят информация в разговор когато някой им би могъл да им каже нещо. Знаем, че когато човек обича, но не е обичан, има мотив да се поддава и би могъл да сподели нещо неволно. Тези и други подобни правила формулираме в езика за правила, който се поддържа от Apache Jena. В таблица 4.2.1 даваме правилата по които наслагваме атрибута „**canShareSensitiveInfoIndirectTo**“ и „**canShareSensitiveInfoTo**“. Тези именно атрибути ще ни помогнат да разберем дали има връзка с хората работещи по секретния проект към други хора.

Правило	Описание
<pre>[r1: (?pX sc:canShareSensitiveInfoIndirectTo ?p1), (?p1 sc:isPartOfConversation ?conv1), (?p2 sc:isPartOfConversation ?conv1), -> (?p1 sc:canShareSensitiveInfoTo ?p2)] [r1a: (?pX sc:canShareSensitiveInfoTo ?p1), (?p1 sc:isPartOfConversation ?conv1), (?p2 sc:isPartOfConversation ?conv1), -> (?p1 sc:canShareSensitiveInfoTo ?p2)]</pre>	Участниците в разговор, които знаят нещо, биха могли да споделят информация
<pre>[r2: (?p1 sc:worksFor sc:roboSoapStrategyProject), (?p2 sc:worksFor sc:roboSoapStrategyProject), -> (?p1 sc:canShareSensitiveInfoTo ?p2), (?p2 sc:canShareSensitiveInfoTo ?p1)]</pre>	Всички работещи по секретния проект биха могли да кажат на някой
<pre>[r3: (?pSomeone sc:canShareSensitiveInfoIndirectTo ?p1), (?p1 sc:isGoodFriendTo ?p2), (?p1 sc:loveButNotLovedProperty ?pX) -> (?p1 sc:canShareSensitiveInfoTo ?p2)]</pre>	Добрите приятели , които не са обичани достатъчно имат мотив да изпуснат информация
<pre>[r4: (?p1 sc:canShareSensitiveInfoTo ?p2), (?p2 sc:canShareSensitiveInfoTo ?p3) -> (?p1 sc:canShareSensitiveInfoIndirectTo ?p3)]</pre>	Транзитивност на споделянето.



[r5: (?p1 sc:canShareSensitiveInfoIndirectTo ?p2), (?p2 sc:canShareSensitiveInfoTo ?p3) -> (?p1 sc:canShareSensitiveInfoIndirectTo ?p3)]	Транзитивност на споделянето.
[r6: (?pSomeone sc:canShareSensitiveInfoIndirectTo ?p2), (?p1 sc:canSeeMonitor ?p2) -> (?p2 sc:canShareSensitiveInfoTo ?p1)]	Всеки, който може да види на монитора на колега нещо би могъл да изпусне информация

Таблица 4.2.1

Правилата се зареждат от файл „src/main/resources/rules.txt“ и се използват по следния начин от Java, дадено в Таблица 4.2.2.

```
public static OntModel makeDeductions(OntModel inputOnt) throws  
IOException {  
    List<Rule> rules = Utils.readRules();  
    // Create reasoner based on our custom rules  
    Reasoner reasoner = new BasicForwardRuleReasoner(rules);  
  
    InfModel inf2 = ModelFactory.createInfModel(reasoner, inputOnt);  
  
    OntModel outOntology =  
    ModelFactory.createOntologyModel(OWL_DL_MEM_RULE_INF, inf2);  
    ...  
}
```

Таблица 4.2.2

Добавяме два нови класа както следва с таблица 4.2.3. С тяхна помощ ще определим пътят на изпуснатата информация и ще оформим в какво се изразява криминалната случка.

LeakInfoSourcePeople \doteq [AND Person [FILLS :worksFor roboSoapStrategyProject] [EXISTS 1 :canShareSensitiveInfoIndirectTo]]
LeakInfoDestinationPeople \doteq [AND Person [FILLS :worksFor domesticSoapSoftwareCompany]]

Таблица 4.2.3

4.3 Изводи със SPARQL^[4]

Вече имаме двете онтологии с индивидите и насложените атрибути, като втората онтология съдържа всички елементи на първата. Сега можем да изпозваме двата нови класа и да направим заключения.

Първо дали има възможно изтичане ще направим със следната заявка от таблица 4.3.1.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX sc: <urn:soap/crime/ontology#>
SELECT ?insider ?outsider
WHERE { ?insider a sc:LeakInfoSourcePeople ;
          sc:canShareSensitiveInfoIndirectTo ?outsider .
          ?outsider a sc:LeakInfoDestinationPeople .
}
```

Таблица 4.3.1

Тук просто потвърждаваме че има възможен теч между хората от LeakInfoSourcePeople към LeakInfoDestinationPeople. Заявката връща три резултата за възможна индиректна (транзитивна) връзка:

1. **insider=john, outsider=sam;**
2. **insider=sara, outsider=sam;**
3. **insider=peter, outsider=sam;**

С втора заявка можем да зададем въпроса към системата къде от секретният проект именно имаме възможен и вероятен пробив. Селектираме всички хора от секретния проект от една страна и всички други възможни хора, които не работят за този проект. Заявката е дадена в Таблица 4.3.2.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX sc: <urn:soap/crime/ontology#>
SELECT ?insider ?outsider
WHERE { ?insider a sc:LeakInfoSourcePeople ;
          sc:canShareSensitiveInfoTo ?outsider .
          MINUS {
            ?outsider sc:worksFor sc:roboSoapStrategyProject
          }
}
```

Таблица 4.3.2

Върнатите резултати са **insider=peter, outsider=riki**. Тоест това е вече криминално деяние и предстои на детектива да установи подробностите.



Третата заявка ни дава хората, между които би могло да изтече информация от фирмата и е дадена в Таблица 4.3.3.

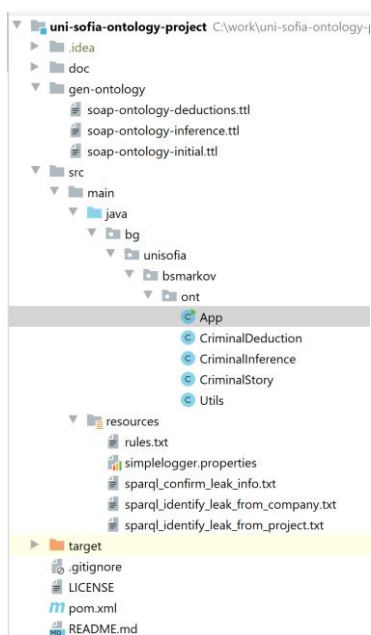
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX sc: <urn:soap/crime/ontology#>
SELECT DISTINCT ?insider ?outsider
WHERE { ?x a sc:LeakInfoSourcePeople ;
          sc:canShareSensitiveInfoIndirectTo ?insider .
          ?insider sc:canShareSensitiveInfoTo ?outsider.
          ?insider sc:worksFor sc:soapAcademyIntCompany.
        MINUS {
          ?outsider sc:worksFor sc:soapAcademyIntCompany
        }
      }
```

Таблица 4.3.3

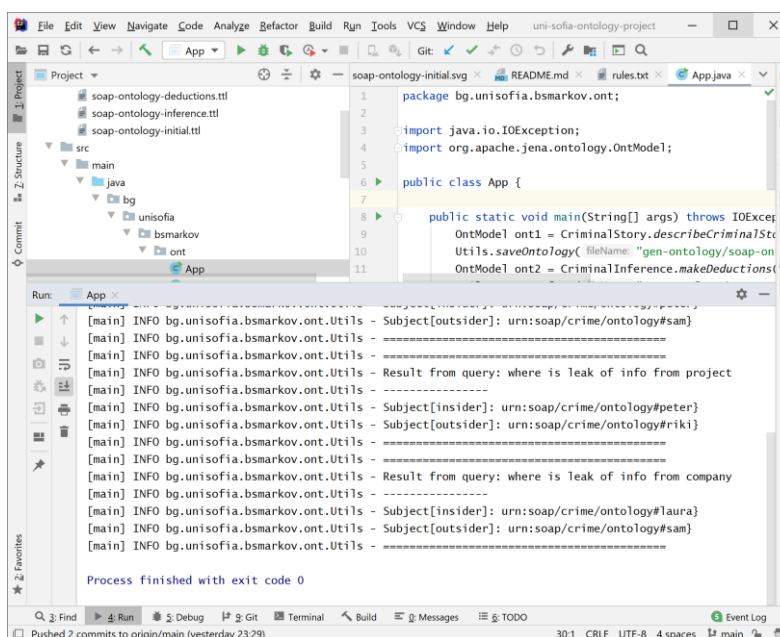
Тук селектираме вече евентуална тройка от хора. А именно – някой който може да сподели информация от проекта във фирмата, друг, който работи във фирмата и му е споделено нещо и трети който му е споделено нещо, но не работи за въпросната фирма. Извеждаме само двамата на границата работещ или не във фирмата soapAcademyIntCompany. Резултатът е **insider=laura**, **outsider=sam**. Това също е криминално деяние и предстои на детектива да установи детайлите.

5. Реализация на проекта

Общата структура на проекта е реализирана с Java, езикът за билдване на проекта е MAVEN. Линк към сорс кода е качен в гитхъб (Вж. Приложение 1) и е неразделна част от този документ. Структурата на приложението е дадена на фигура 5.1. Резултатът от едно изпълнение е даден на фигура в 5.2. Използваната развойна среда е IntelliJ.



Фигура 5.1



Фигура 5.2

Главният изпълним файл е App.java и в него се извикват статични методи от останалите класове. Подробни инструкции са дадени в README.md файла. Например първоначално класовете и инстанциите се създават по следния начин:

```
OntModel ont1 = CriminalStory.describeCriminalStory();
```

Втората стъпка е правенето на изводи и вторична онтология:

```
OntModel ont2 = CriminalInference.makeDeductions(ont1);
```

Последната стъпка е извикване на SPARQL заявки и печат на конзолата:

```
CriminalDeduction.criminalDeduction(ont2);
```

6. Идеи за бъдещо развитие и подобрения

Изложеният пример на криминална случка е само една демонстрация на технологията в един нейн вариант. Би могло проекта да се адаптира към по-общ модел на данни, вкарвани или извличани по автоматизиран начин. Например някакви взаимодействия на хора от Фейсбук, или трафик на автомобили с цел проследимост. Би могло също така



описателната част (онтологията) да е статична, а индивидите или самите данни да се събират постоянно от Интернет или от някакви датчици.

7. Източници и използвана литература

[1] Knowledge Representation and Reasoning, Ronald Brachman, Hector Levesque, 2004, The Morgan Kaufmann Series in Artificial Intelligence

[2] Apache Jena project,

<https://jena.apache.org/documentation/inference/index.html#OWLconfiguration>

[3] Web Ontology Language (OWL), W3C, <https://www.w3.org/OWL/>

[4] SPARQL By Example, W3C, <https://www.w3.org/2009/Talks/0615-qbe/>

Приложения

1. Сурс код (Source code)

<https://github.com/borkox/uni-sofia-ontology-project>