

Софийски университет “Св. Климент  
Охридски”

Факултет по математика и информатика

# Проект

по

“Представяне и моделиране  
на знания”

на тема

“Онтология за Yu-Gi-Oh! TCG”

Изготвен от:

Иван Арабаджийски, ФН: 5MI3400052,  
спец. Извличане на информация и обработка на знания

1.02.2022

# Съдържание

1. Идея на проекта.....	3
2. Елементи на онтологията.....	3
2.1 Класове.....	3
2.2 Свойства.....	4
2.2.1 Свойства на обектите.....	4
2.2.2 Свойства на данните.....	4
2.3 Индивиди.....	5
3. Примери за извърпване на логически извод.....	6
4. Извършване на класификация.....	6
4.1 Правила.....	6
5. Заявки към базата от знание.....	6
5.1 SPARQL заявки.....	6
6. Схема на онтологията.....	8
7. Използвани технологии.....	8

# 1. Идея на проекта

Идеята на проекта е да се изгради онтология в предметната област на

играта с карти Yu-Gi-Oh! TCG (trading card game). Тук включваме картите за игра, тяхната класификация, обединението им в тестета, играчите, които играят играта, участието им в турнири и печалбите. Крайните цели са:

- да се улесни класифицирането на новодобавени карти (все пак нови карти излизат на всеки 3-4 месеца)
- по-бързо намиране на турнир, неговите участници, победители, мачове и прочие.
- търсене на специфични карти по техните свойства

## 2. Елементи на онтологията

### 2.1 Класове

Основните класове, с които работим са Card – картите биват три вида: Monster, Spell и Trap. Спрямо някои свои основни характеристики чудовищата се разпределят в още подкласове.

Monster  $\sqsubseteq$  Card

Spell  $\sqsubseteq$  Card

Trap  $\sqsubseteq$  Card

NonEffectMonster  $\doteq$  [AND Monster  
[ $\neg$ FILLS :has\_effect Effect]]

EffectMonster  $\doteq$  [AND Monster  
[FILLS :has\_effect Effect]]

NonEffectMonster  $\doteq$  [AND Monster  
[EXISTS 2 :has\_effect]]

XyzMonster  $\doteq$  [AND Monster  
[FILLS :has\_rank integer]]

LinkMonster  $\doteq$  [AND Monster  
[FILLS :has\_link\_rating integer]]

FusionMonster  $\sqsubseteq$  Monster

SynchroMonster  $\sqsubseteq$  Monster

RitualMonster  $\sqsubseteq$  Monster

NormalMonster  $\sqsubseteq$  NonEffectMonster

Attribute  $\doteq$  {Dark}  $\sqcup$  {White}  $\sqcup$  {Water}  $\sqcup$  {Earth}  $\sqcup$  {Fire}  $\sqcup$  {Wind}  $\sqcup$  {Divine}

Type  $\doteq$  {Machine}  $\sqcup$  {Fairy}  $\sqcup$  {Dragon}  $\sqcup$  {Warrior}  $\sqcup$  {Zombie}  $\sqcup$  {Spellcaster}  $\sqcup$  {Winged-Beast}

Ability  $\doteq$  {Union}  $\sqcup$  {Tuner}  $\sqcup$  {Flip}  $\sqcup$  {Gemini}  $\sqcup$  {Toon}  $\sqcup$  {Spirit}

Spell и Trap картите имат свои подразделения

QuickPlaySpell  $\sqsubseteq$  Spell

RitualSpell  $\sqsubseteq$  Spell

NormalSpell  $\sqsubseteq$  Spell

EquipSpell  $\sqsubseteq$  Spell

ContiniousSpell  $\sqsubseteq$  Spell

FieldSpell  $\sqsubseteq$  Spell

ContiniousTrap  $\sqsubseteq$  Trap

CounterTrap  $\sqsubseteq$  Trap

NormalTrap  $\sqsubseteq$  Trap

Следващите основни класове върху, които да се концентрираме са Player, Deck, Match и Tournament. Имайки опит играчите могат да бъдат ExperiencedPlayer и CrowdFavourite. Когато хората си избират фаворит обикновено гледат хората с най-много опит. ExperiencedPlayer са играчи с поне 7 години опит, а фаворитите са играчи с поне 5 години опит.

Draw  $\doteq$  [AND Match  
[ $\neg$ FILLS :has\_winner]]

## 2.2 Свойства

### 2.2.1 Свойства на обектите

Основните свойства са за определяне на картите, както и играчите в турнири.

Domain	Property	Range	Characteristics
Card	has_effect	Effect	exactly(2)
Monster	has_type	Type	functional
Monster	has_ability	Ability	
Monster	has_attribute	Attribute	functional
Tournament	has_rounds	Round	
Round	has_matches	Match	
Match	has_players	Player	inverseOf plays_in_match
Match	has_winner	Player	
Card	is_used_in	Deck	inverseOf has_cards
Deck	has_cards	Card	inverseOf is_used_in
Player	participates_in	Tournament	
Player	plays_in_match	Match	inverseOf has_players

### 2.2.2 Свойства на данните

Голяма част от свойствата на картите са прости данни. Чудовищата имат атака, защита, ниво (числа), както и ефект, име (текст). Играчите, които играят в турнири, имат id номер, точки, които са важни за крайното класиране и тн.

Domain	Property	Range	Characteristics
Effect	has_text	string	functional
Monster	attack	integer	functional
Monster	defense	integer	functional
Monster	level	integer	functional
Monster	rank	integer	functional
Monster	link_rating	integer	functional
Card	name	string	functional
Card	rarity	OneOf(['Common', 'Rare', 'Super Rare', 'Ultra Rare', 'Ultimate Rare', 'Ghost Rare'])	functional
Deck	archtype	string	
Player	has_points	integer	functional
Player	number_of_wins	integer	functional
Player	number_of_draws	integer	functional
Player	years_experience	integer	functional

## 2.3 Индивиди

Картите в играта са над 10000 така че тук показваме само по няколко индивида от клас за нагледност.

Ivan → [AND Player

[FILLS :years\_experience 8]  
[FILLS :plays\_in\_match Round1Match1]  
[FILLS :plays\_in\_match Round2Match1]  
[FILLS :uses SkyStrikerDeck]  
[FILLS :has\_points 6]]

Dark → Attribute

{Dark} ≠ {Light} ≠ {Water} ≠ {Earth} ≠ {Wind} ≠ {Fire} ≠ {Divine-Beast}

UtopiaEffect → [AND Effect

[FILLS :has\_text “When a monster declares an attack: You can detach 1 material from this card; negate the attack. If this card is targeted for an attack, while it has no material: Destroy this card.”]]

Number 39: Utopia → [AND Monster

[FILLS :effect UtopiaEffect]  
[FILLS :rank 4]  
[FILLS :atk 2500]  
[FILLS :defense 2000]  
[FILLS :has\_attribute Light]  
[FILLS :has\_type Warrior]]

Round1Match1 → [AND Match

[FILLS :has\_players Ivan]  
[FILLS :has\_players Ivaylo]

```

Round2Match2 → [AND Match
                  [FILLS :has_winner Ivan]]
                  [FILLS :has_players Ventseslav]
                  [FILLS :has_players Kostadin]]

```

### 3. Примери за извърпване на логически извод

KB |= (Ivan → CrowdFavourite)

В случая Ivan има 8 години опит, а фаворитита е играч с над 7.

KB |= (Round2Match2 → Draw)

Този мач няма победител, следователно той е равенство.

KB |= (Number39:Utopia → EffectMonster)

KB |= (Number39:Utopia → XyzMonster)

Чудовището Number39:Utopia има ранк и ефект, тоест то е едновременно EffectMonster и XyzMonster.

### 4. Извършване на класификация

ExperiencedPlayer е играч с поне 7 години опит, а CrowdFavourite е такъв с поне 5 години опит. Оттам следва, че всеки ExperiencedPlayer е и CrowdFavourite.

ExperiencedPlayer → CrowdFavourite → Player → Person → Thing

#### 4.1 Правила

Пример за правило, което олеснява класификацията. Автоматичното присвояване на турнири, в които са играли хората. Ако човек е участвал в мач от даден турнир, то той е участвал в турнира.

```

Player(?p), plays_in_match(?p, ?m), has_matches(?r, ?m), has_rounds(?t, ?r) ->
participates_in(?p, ?t)

```

### 5. Заявки към базата от знание

#### 5.1 SPARQL заявки

- Името и броят на картите във всяко тесте

```

SELECT ?deck (COUNT(DISTINCT ?card) AS ?count)
WHERE {

```

```

        ?deck yugiOntology:has_cards ?card .
    }
GROUP BY ?deck

```

```

result = [[yugiOntology.SkyStriker, 3], [yugiOntology.Odd-Eyes, 1],
          [yugiOntology.Eldlich, 3]]

```

- Играчът с най-много точки

```

SELECT ?player ?p ?d
WHERE {
    ?player yugiOntology:has_points ?p .
    ?player yugiOntology:uses ?d
}
ORDER BY desc(?p)
LIMIT 1

```

```

result = [[yugiOntology.Ivan, 6, yugiOntology.SkyStriker]]

```

- Всички чудовища с ефект

```

SELECT ?card
WHERE {
    ?card rdf:type yugiOntology:EffectMonster .
}

```

```

result = [[yugiOntology.SkyStrikerAce-Raye], [yugiOntology.AshBlossom&JoyousSpring],
          [yugiOntology.Number39:Utopia], [yugiOntology.CrystonHalqifibrax]]

```

- Всеки победител в турнир и техните тестета

```

SELECT ?player ?p ?d ?tournament
WHERE {
    ?player yugiOntology:has_points ?p .
    ?player yugiOntology:uses ?d
    ?player yugiOntology:participates_in ?tournament
}
GROUP BY ?tournament
ORDER BY desc(?p)

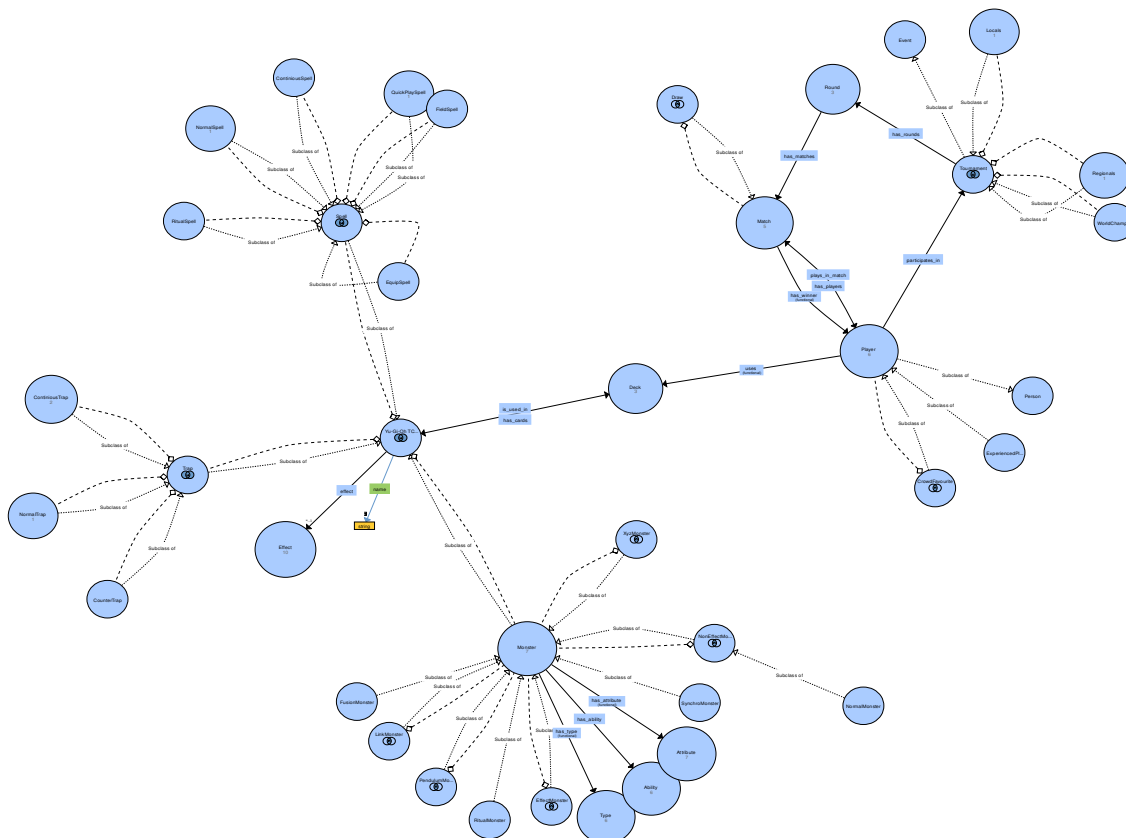
```

```

result = [[yugiOntology.Ivan, 6, yugiOntology.SkyStriker,
          yugiOntology.SofiaRegionals30.11.2022],
          [yugiOntology.Pesho, 3, yugiOntology.Eldlich,
          yugiOntology.PlovdivLocals06.02.2022]]

```

## 6. Схема на онтологията



## 7. Използвани технологии

- [Python3](#) - Проектът е реализиран на езика Python
- [Owlready2](#) - Фреймуърк за Python, ориентиран към работа с онтологии.
- За визуализиране схемата на онтологията е използван инструментът [WebVOWL](#)

[Documentation](#) | [Project Repository](#)