



СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

КУРСОВ ПРОЕКТ
ПО
ПРЕДСТАВЯНЕ И МОДЕЛИРАНЕ НА ЗНАНИЯ

Тема:

Онтология за софтуер в областта на науката за данни
(Data Science Software Ontology - DSSO)

Студент:

Семир Балджиев, 0MI3400731, ИИОЗ

София, януари 2026 г.

1. Формулировка на задачата

1.1. Мотивация

Софтуерът в областта на науката за данни се характеризира с голям растеж на инструменти, библиотеки, алгоритми имплементирани в тях и хардуерни ресурси с които софтуерът може да работи. Липсата на единен формализиран модел за описание на тези ресурси затруднява автоматизираното и семантично откриване на съвместимост, проследяване на зависимостите и класификация на софтуера спрямо възможностите които предоставя и характеристиките му.

1.2. Цел

Целта е да се разработи онтология базирана на OWL 2 DL, която да формализира знанията за съществуващия софтуер в областта на науката за данни. Онтологията трябва да позволява автоматично разсъждения и откриване на скрити зависимости.

1.3. Дефиниране на задачи за постигане на целта

1. **Изграждане на тааксономия:** Дефиниране на йерархична структура от класове, покриващи основата на предметната област.
2. **Моделиране на връзките:**
 - a. Дефиниране на обектни свойства с основни логически характеристики за *транзитивност, симетрия, инверсия и функционалност*.
 - b. Дефиниране на свойства за данни които описват атрибутите на класовете, чрез задаване на примитивен тип данни за стойност на атрибута.
3. **Логическо моделиране:** Създаване на дефинирани класове, които използват логически оператори за (*AND, OR, NOT*) и квантори за *всеобщност и съществуване*. Целта е да се композират по-комплексни класове с цел автоматично разпознаване на категории като например (*Open Source Software*), на базата на техните свойства, а не ръчно да се аотират към съответните категории.
4. **Създаване на индивиди с реални данни:** Въвеждане на данни от представителна извадка за предметната област, които да демонстрират възможностите на онтологията.
5. **Валидация и разсъждения/изводи:** Използването на програма, която извършва разсъждения върху онтологията за проверка на логическата

състоятелност. Генериране на нови знания и изводи относно индивидите и техните свойства.

1.4. Съществуващи решения

В тази предметна област на формално описание и семантично моделиране на софтуер съществуват няколко утвърдени онтологии и стандарти. Въпреки тяхната значимост те са твърде общи или тясно специфични към конкретна предметна област (напр. Биоинформатиката). По това се различава и текущия проект, тъй като е в областта на науката за данни, в която липсват формулирани бази от знания и онтологии, които да бъдат в полза на различни специалисти. Този проект е взаимстван от съществуващите и ги надгражда чрез дефиниране на специфични класове и свойства за предметната област които не са общи за софтуерни артефакти, както са в съществуващите.

1. The Software Ontology (SWO)

- а. Това е една от най-изчерпателните онтологии за описване на софтуерни инструменти, първоначално разработена за биомедицинската сфера. Структурата е изключително сложна и тежка за конкретните нужди на науката за данни, като липсват специфични концепции за тази област.

2. Machine Learning Schema

- а. Това е лека онтология, предназначена за описване на експерименти с машинно обучение. Тя се фокусира върху процеса на изпълнение на ML модели (runs), хиперпараметри и метрики за оценка

2. Описание на онтологията

2.1 Йерархия на класовете

Клас	Родител	Описание
Software	Thing	Основен клас за всички софтуерни артефакти
DataScienceSoftware	Software	Софтуер, специализиран за задачи в науката за данни.

MLLibrary	DataScienceSoftware	Библиотеки за машинно обучение (напр. TensorFlow, PyTorch).
DataVisualizatioTool	DataScienceSoftware	Инструменти за визуализация на данни (напр. Tableau, Matplotlib).
NotebookEnvironment	DataScienceSoftware	Интерактивни среди за разработка (напр. Jupyter).
CloudPlatform	DataScienceSoftware	Облачни платформи за изчисления (напр. Databricks).
Algorithm	Thing	Абстрактен клас за математически алгоритми.
MachineLearningMethod	Algorithm	Методи за машинно обучение.
SupervisedLearningMethod	MachineLearningMethod	Алгоритми за обучение с учител (напр. Регресия).
UnsupervisedLearningMethod	MachineLearningMethod	Алгоритми за обучение без учител (напр. Клъстеризация)
ReinforcementLearningMethod	MachineLearningMethod	Алгоритми за обучение на принципа подсилено обучение
DeepLearningMethod	MachineLearningMethod	Дефиниран клас: Автоматично се дефинира, ако алгоритъмът използва невронна мрежа.
OptimizationAlgorithm	Algorithm	Алгоритми за оптимизация (напр. Gradient Descent).
ModelArchitecture	Thing	Структура/Архитектура на модела.
NeuralNetwork	ModelArchitecture	Клас за всички видове невронни мрежи.
FeedForward	NeuralNetwork	Стандартни (права връзка) мрежи.
Recurrent	NeuralNetwork	Рекурентни невронни мрежи (RNN).
Convolutional	NeuralNetwork	Конволюционни мрежи (CNN).

Transformer	NeuralNetwork	Трансформър архитектури (напр. BERT, GPT).
HardwareResource	Thing	Физически ресурси за изпълнение.
HardwareAccelerator	HardwareResource	Ускорители за изчисления (GPU, TPU).
InstructionSet	HardwareResource	Набори от инструкции (напр. x86, ARM).
License	Thing	Правни лицензи за софтуера.
OpenSourceLicense	License	Лицензи с отворен код (MIT, Apache 2.0).
ProprietaryLicense	License	Затворени/Комерсиални лицензи.
Organization	Thing	Организации, разработващи софтуера.
OperatingSystem	Thing	Операционни системи с които е съвместим софтуера (Linux, Windows).
ProgrammingLanguage	Thing	Езици за програмиране на които е написан софтуера (Python др.)
DataFormat	Thing	Формати на входните данни (CSV, Parquet).
Metric	Thing	Метрики за оценка които могат да се използват от различните алгоритми (Accuracy, F1 Score).

2.2. Описание на свойствата

Обектни свойства				
Свойство	Domain (От кой клас)	Range (Към кой клас)	Логическа характеристика	Описание

Обектни свойства				
depends_on	Software	Software	Transitive	Основна връзка за зависимости. Тъй като е транзитивна , ако А зависи от В, а В зависи от С, системата автоматично извежда, че А зависи от С.
requires_dependency	Software	Software	SubProperty на depends_on	По-специфична под-връзка на depends_on.
interoperates_with	Software	Software	Symmetric	Показва съвместимост. Тъй като е симетрична , ако дефинираш PyTorch <-> Numpy, връзката автоматично става двупосочна.
has_license	Software	License	Functional	Указва лиценз на софтуера. Тъй като е функционална , един софтуер може да има само един <i>основен</i> лиценз в този контекст.
implements_algorithm	Software	Algorithm	Inverse на is_implemented_by	Връзка от инструмента към математическия метод (напр. Scikit-Learn имплементира Random Forest)
is_implemented_by	Algorithm	Software	Inverse на implements_algorithm	Обратната връзка. Позволява да питаш „Кои инструменти реализират този алгоритъм?“.
developed_by	Software	Organization	Standard	Свързва софтуера с неговия създател или организацията, която го поддържа.
written_in	Software	ProgrammingLangu	Standard	Указва езиците за

Обектни свойства				
		age		програмиране, използвани за написването на инструмента.
runs_on_os	Software	OperatingSystem	Standard	Дефинира съвместимостта с операционни системи (Linux, Windows и др.).
uses_architecture	Algorithm	ModelArchitecture	Standard	Свързва алгоритъм (напр. GPT) с архитектурата на невронната мрежа (напр. Transformer). Ключова за дефиниране на Deep Learning.
accelerated_by	Software	HardwareAccelerator	Standard	Указва дали софтуерът може да използва GPU/TPU ускорение.
supports_format	Software	DataFormat	Standard	Какви файлови формати (CSV, Parquet) може да чете/пише инструментът.
optimizes_metric	Algorithm	Metric	Standard	Използвани метрики

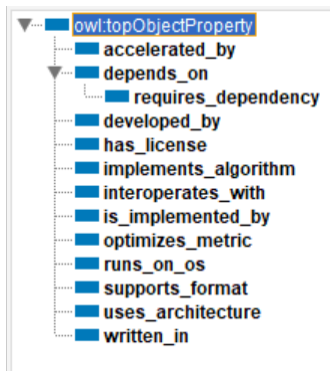
Свойства за данните				
Свойство	Domain	Range	Характеристика	Описание
released_in_year	Software	int	Standard	Годината на официалното пускане на софтуера. Използва се за филтриране (напр. софтуер след 2020 г.).
is_deprecated	Software	bool	Functional	Флаг, указващ дали софтуерът е остарял и вече не се поддържа.
has_version	Software	str	Functional	Текстово представяне на текущата стабилна версия

Свойства за данните				
				(напр. "2.1.0").

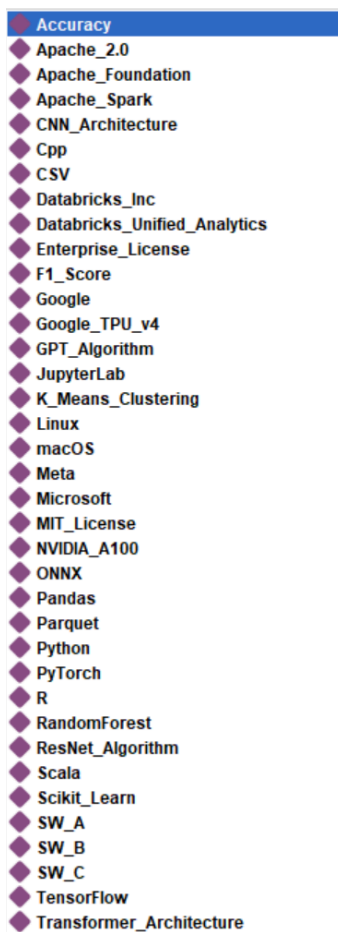
2.3. Описание на дефинираните класове (Defined classes)

Дефиниран Клас	Описание на Логиката	Тип на ограничението
DeepLearningMethod	Метод за машинно обучение, който използва архитектура на невронна мрежа.	Existential Restriction (Intersection/AND)
DeepLearningSoftware	Софтуерен инструмент, който имплементира поне един алгоритъм, класифициран като DeepLearningMethod.	Property Chain Inference (Верижна инференция)
ApacheProject	Софтуер, който е разработен конкретно от организацията "Apache Foundation".	Value Restriction (Nominal / has_value)
EnterprisePlatform	Софтуер, разработен от голяма технологична корпорация (Google, Microsoft ИЛИ Databricks).	Union Class (Disjunction/OR)
PurePythonTool	Софтуерен инструмент, който е написан на езика за програмиране Python.	Value Restriction (Existential)

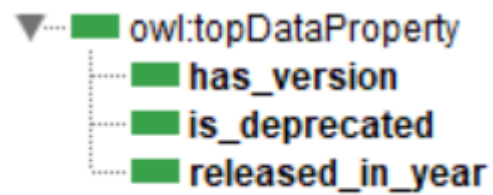
3. Визуализации



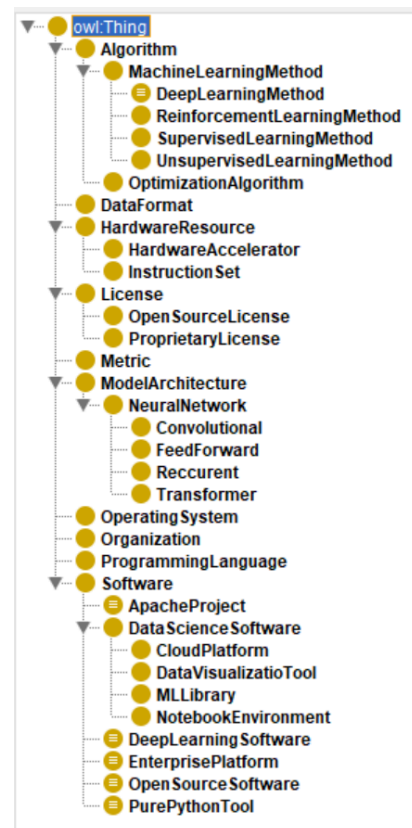
Фиг. 1. Обектни свойства



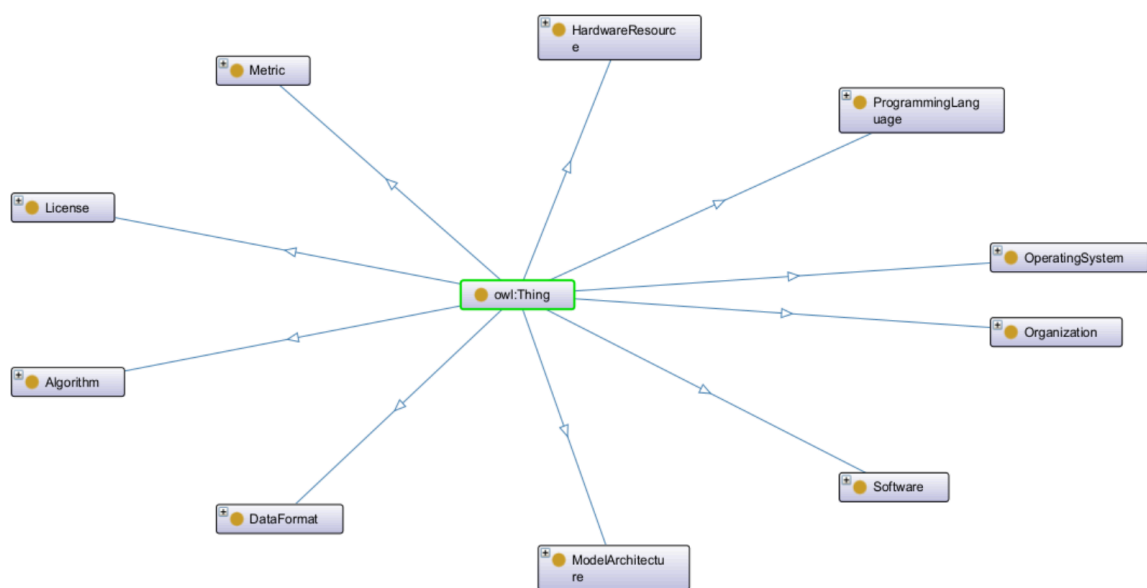
Фиг. 3. Индивиди



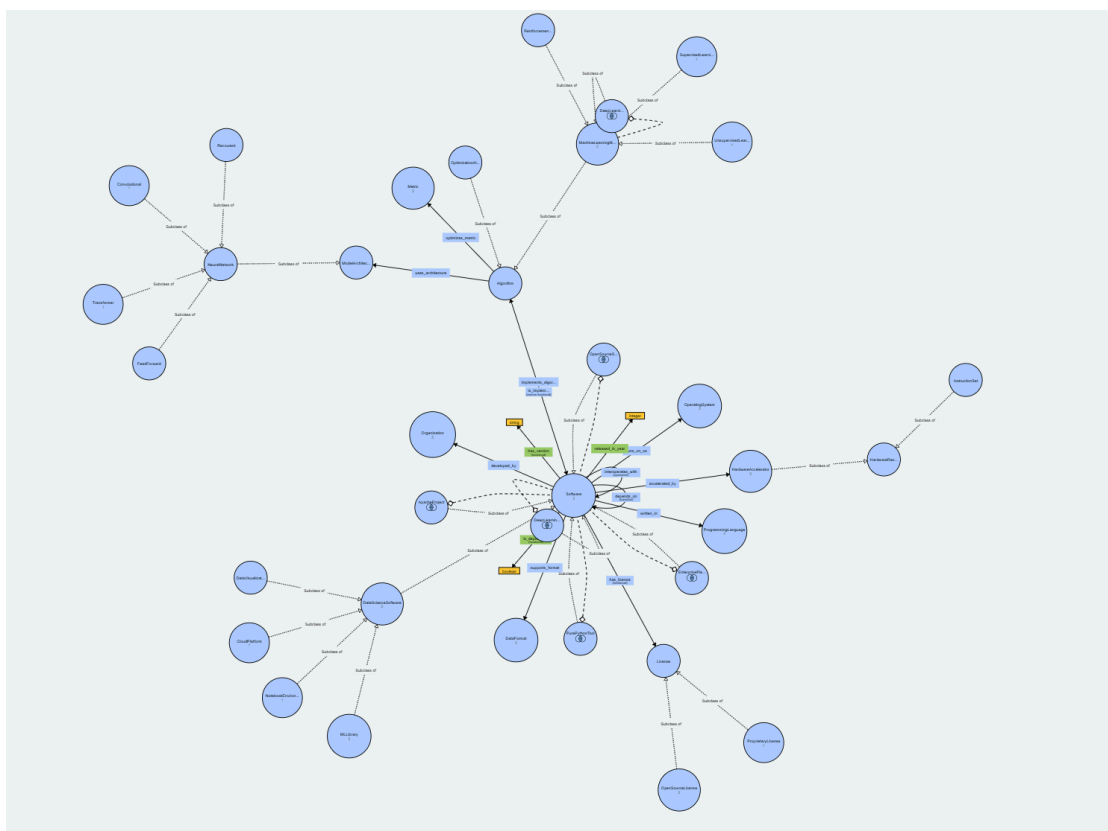
Фиг. 2. Свойства за данни



Фиг. 4. Йерархия от класове



Фиг. 5. Визуализация на първото ниво на онтологията с Protege



Фиг. 6. Визуализация на цялостната онтология с WebVOWL

4. Примерни DL заявки

Query (class expression)

Software and implements_algorithm some (uses_architecture some NeuralNetwork)

Query results

Equivalent classes (0 of 0)

Superclasses (1 of 2)

- Software

Direct superclasses (1 of 1)

- Software

Direct subclasses (1 of 1)

- ☰ DeepLearningSoftware

Subclasses (1 of 2)

- ☰ DeepLearningSoftware

Instances (3 of 3)

- ◆ PyTorch
- ◆ SW_A
- ◆ TensorFlow

Фиг. 7. Примерна заявка която извежда всички софтуери които използват NN архитектура

Query (class expression)

Software and (developed_by value Google or developed_by value Meta)

Query results

Equivalent classes (0 of 0)

Superclasses (1 of 2)

- Software

Direct superclasses (1 of 1)

- Software

Direct subclasses (0 of 1)

Subclasses (0 of 1)

Instances (2 of 2)

- ◆ PyTorch
- ◆ TensorFlow

Фиг. 8. Намира софтуер разработен от Google или Meta

5. ИСТОЧНИЦИ

1. Malone, J., Brown, A., Lister, A. L., Ison, J., Hull, D., Parkinson, H., & Stevens, R. (2014). The Software Ontology (SWO): a resource for reproducibility in biomedical data analysis, curation and digital preservation. *Journal of biomedical semantics*, 5, 25. <https://doi.org/10.1186/2041-1480-5-25>
2. [ML Schema Core Specification](#)
3. W3C OWL Working Group. [OWL 2 Web Ontology Language: Overview](#). 11 December 2012
4. Protege
5. Lamy JB. [Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies](#). *Artificial Intelligence In Medicine* 2017;80:11-28