CryptOntology

Semantic web e Criptovalute.

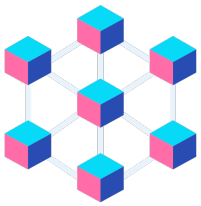
Alessandro Mini - 7060381
Andrea Neri - 7060638
Silviu Robert Plesoiu - 7051276

Elaborato del corso di Data Warehousing
2020-2021

C.D.L Magistrale di Ingegneria Informatica
Università degli Studi di Firenze

Indice

1 Introduzione



CryptOntology è un progetto per la costruzione di un sistema informativo (Ontologia) per accedere a informazioni relative alle principali *criptovalute* attualmente disponibili nel mercato.

Le informazioni disponibili sono principalmente di carattere tecnico, tecnologie di implementazione, possibilità di eseguire smart contract etc.

L'utilità di questo sistema consiste nel fatto che non vengono raccolte informazioni legate all'aspetto prettamente economico e finanziario delle criptovalute ma si vuole costruire una *knowledge base* per raccogliere e classificare varie tecnologie.

Nell'implementazione dell'ontologia verrà particolarmente curato il rapporto tra le varie classi e data properties, sarà inoltre presente un integrazione con *Dbpedia* per reperire ulteriori informazioni sui vai *building blocks* delle criptovalute.

2 Narrazione

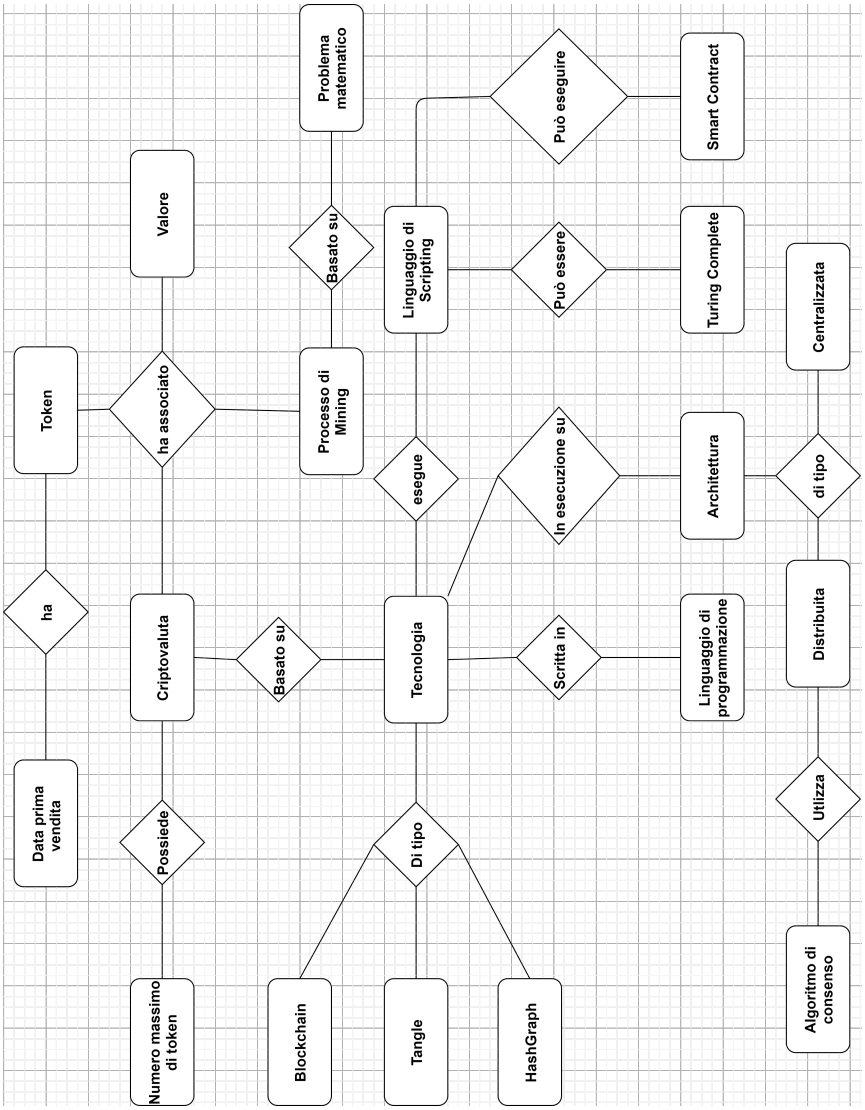
Per implementare l'ontologia si è studiato approfonditamente l'argomento e le tecnologie di fondo delle criptovalute, arrivando a redigere la seguente *narrazione*:

Una **criptovaluta** è un tipo di moneta digitalizzata a cui è associato un valore e basata su una certa tecnologia (Blockchain, Tangle, hashgraph).
Ad una criptovaluta può essere associato un processo di **mining** generalmente basato sulla risoluzione di un problema matematico, come l'utilizzo di funzioni hash in Bitcoin.
Ogni criptovaluta ha associato un **Token** (es. Bitcoin: BTCT) a cui è legata una data della prima vendita e una criptovaluta possiede un campo che determina il numero massimo di token disponibili (es. Token 21M).
Una **tecnologia** esegue un linguaggio di scripting che può essere o meno turing completo e in cui possono eventualmente essere eseguiti gli smart contracts (se la tecnologia lo permette).
Ogni tecnologia di fondo può essere eseguita su un **architettura** di tipo distribuita o centralizzata e nel caso in cui sia distribuita è necessaria la presenza di un algoritmo di consenso (Proof Of work/Stake/Importance, Raft ...).

In cui emergono le **keyword**: criptovaluta, tecnologia, blockchain, tangle, hashgraph, accesso pubblico, accesso privati, tecnologia di fondo, architettura centralizzata, distribuita, linguaggio di scripting (più o meno turing completo), smart contracts, mining, token, algoritmo di consenso.

2.1 Diagramma ER

Dalla narrazione si è costruito il *diagramma ER*:



3 Implementazione

Per implementare l'ontologia viene sfruttato il tool Protegè, un programma che raccoglie una serie di strumenti utili per la rappresentazione della conoscenza sviluppato dall'università di Stanford (https://protege.stanford.edu/), l'implementazione prevede, in ordine:

1. Classi.
2. Object properties.
3. Data properties.
4. Individuals.

3.1 Implementazione delle Classi

Nel progetto sono state implementate le seguenti classi:

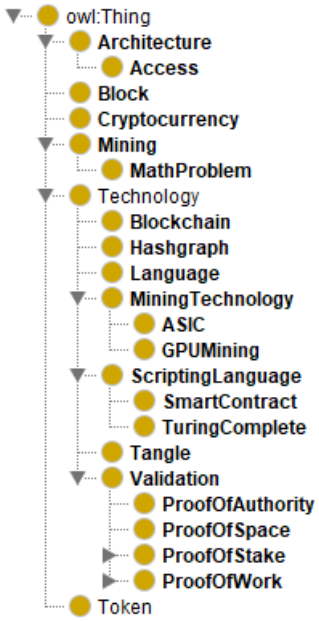


Figura 1: Classi implementate nel progetto

Si può notare che le classi sottostanno agli elementi principali della narrazione, il loro rapporto gerarchico viene specificato in Figura 2.

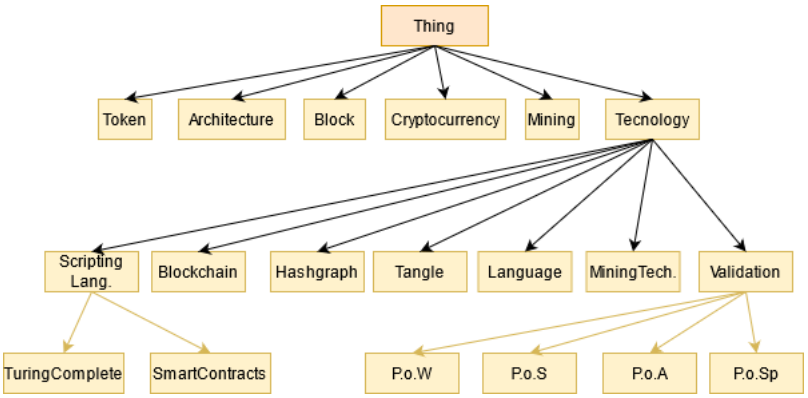


Figura 2: Diagramma delle classi e rapporti parent/children.

Se ne riporta brevemente il significato ed eventuali considerazioni:

1. **Architecture**: Prevede l'architettura di fondo della criptovaluta, ad esempio se questa è centralizzata o distribuita.

2. **Block**: Rappresenta la classe relativa al blocco.

3. **CryptoCurrency**: Classe relativa alla criptovaluta.

4. **Mining**: Rappresenta il processo di Mining, ha come children la classe *MathProblem* Che rappresenta il problema matematico su cui è basato il mining.

5. **Technology**: E' la tecnologia di fondo su cui si basa la criptovaluta, ad esempio Bitcoin si basa sulla BitcoinBlockchain, le classi Technology e Architecture potrebbero sembrare simili ma ricoprono ruoli logicamente differenti.

   Una *tecnologia* (come la blockchain) può essere eseguita in un *architettura* distribuita oppure, anche se forzatamente, su una centralizzata.

   Questo ragionamento può essere esteso anche ai metodi di accesso, in quest'ottica quindi la tecnologia è intesa come il "motore" che permette il funzionamento della criptovaluta mentre l'architettura può essere vista come la struttura di deployment.

6. **Blockchain,Hashgraph,Tangle:** Sono classi che rappresentano tre tecnologie principali che ricorrono in tutte le criptovalute analizzate.

7. **Language**: E' la classe che rappresenta il linguaggio di programmazione della criptovaluta.

   Anche in questo caso è opportuno eseguire una disambiguazione con la classe ScriptingLanguage (linguaggio di scripting) in quanto potrebbe sembrare che siano effettivamente la stessa cosa.

Con *Language* si intende il linguaggio di programmazione dell'infrastruttura della criptovaluta, per Bitcoin questo è il C++ [1].

Un *linguaggio di scripting* è un linguaggio ad altissimo livello costruito sopra l'architettura per eseguire smart contracts e simili, per Bitcoin questo linguaggio è lo Script [2].

8. **MiningTechnology** Specifica il mining all'interno della tecnologia di implementazione, e quindi se è preferibile un processo di tipo ASIC oppuire l'utilizzo di GPU. Questo parametro sarà OPTIONAL per molte criptovalute in quanto le più moderne sovvertono il classico concetto di mining, in implementazioni come Proof of stake.

9. **Validation** La classe Validation descrive l'algoritmo di consenso utilizzato, sempre presente nelle criptovalute analizzate, prevede 4 figli:

   (a) Proof of Work.

   (b) Proof of Space.

   (c) Proof of Stake.

   (d) Proof of Authority.

3.2 Implementazione delle Object Properties

Si sono implementate le seguenti Object Properties, ovvero delle relazioni che legano due individuals attraverso un predicato:

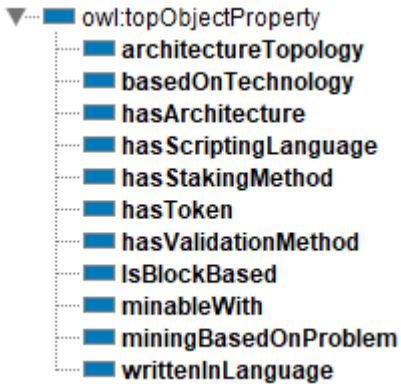


Figura 3: Object properties implementate.

In cui i domini e codomini sono i seguenti:

1. **architectureTopology** : Cryptocurrency $\longrightarrow$ Architecture.

2. **basedOnTechnology** : Cryptocurrency $\longrightarrow$ Technology.

3. **hasArchitecture** : Technology ⟶ Architecture.
4. **hasScriptingLanguage** : Technology ⟶ ScriptingLanguage.
5. **hasToken**: CryptoCurrency ⟶ Token.
6. **hasValidationMethod**: Technology ⟶ Validation.
7. **MinableWith**: CryptoCurrency ⟶ MiningTechnology.
8. **MiningBasedOnProblem**: MiningTechnology ⟶ MathProblem.
9. **WrittenInLanguage**: CryptoCurrency ⟶ Language.

3.3 Implementazione delle Data Properties

Si sono implementate le seguenti Data Properties, delle proprietà degli individuals che possono essere identitifcate come attributi:

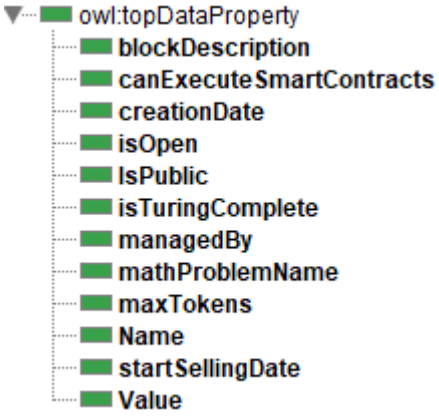


Figura 4: Data properties implementate.

Corrispondenti ai seguenti tipi:

1. **blockDescription** : Block ⟶ String.
2. **canExecuteSmartContracts** : ScriptingLanguage ⟶ boolean.
3. **creationDate** : CryptoCurrency ⟶ dateTime.
4. **isOpen** : Access ⟶ boolean.
5. **isClosed**: Architecture ⟶ boolean.
6. **isPublic**: Architecture ⟶ boolean.
7. **isTuringComplete**: ScriptingLanguage ⟶ boolean.

8. **managedBy**: CryptoCurrency ⟶ String.
9. **mathProblemName**: Mathproblem ⟶ String.
10. **maxTokens**: CryptoCurrency ⟶ String.
11. **Name**: CryptoCurrency ⟶ String.
12. **startSellingDate**: Token ⟶ dateTime.
13. **maxTokens**: CryptoCurrency ⟶ float.

3.4 Implementazione degli individuals

Come ultima fase si sono implementate 4 criptovalute per eseguire dei test e delle query, dato l'elevato numero di classi e proprietà inserire criptovalute è un operazione onerosa.

Si sono implementate:

1. Bitcoin.
2. Monero.
3. Cardano.
4. Ethereum.

Le quali sono rilevanti in quanto sottostanno a principi molto diversi.

A queste criptovalute corrispondono i seguenti individuals:

Figura 5: Individuals inseriti nell'ontologia corrispondenti alle 4 criptovalute.

Un esempio di collegamento Tra i vari oggetti, cioè di utilizzo di object properties e data properties viene riportato di seguito ed è riferito alla criptovaluta "Bitcoin":

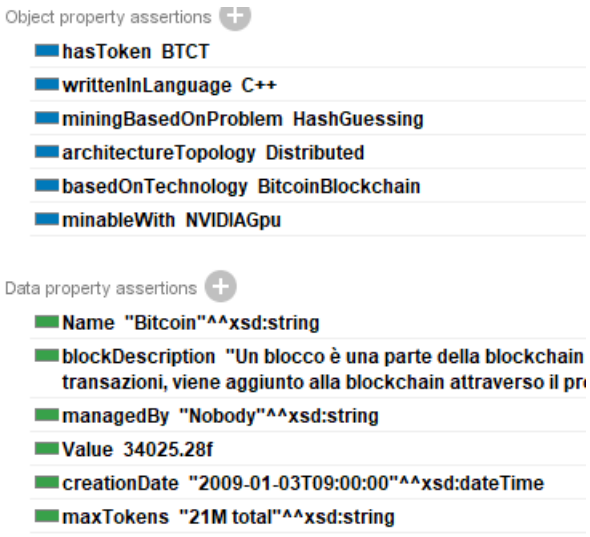


Figura 6: Relazioni associati all'individual "Bitcoin".

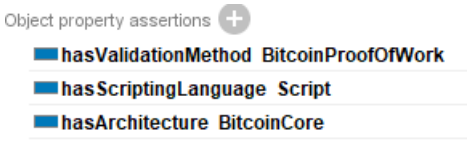


Figura 7: Relazioni associati all'individual "BitcoinBlockchain".

Figura 8: Relazioni associati all'atchitettura di Bitcoin , ovvero "BitcoinCore".

Questi dati costituiscono gli elementi principali dell'architettura di bitcoin così come è stata implementata nell'ontologia, sono presenti anche altri elementi come lo scripting language ("Script") e il token.

4 Query SPARQL

Al fine di testare la funzionalità del sistema informativo sono state testate alcune query, si utilizzeranno i seguenti prefissi:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX mine:
↪ <http://www.semanticweb.org/alessandro/ontologies/2021/4/ProgettoDW#>
```

Le query testate sono le seguenti:

1) Recuperare le criptovalute con valore al momento della creazione superiore a 25.000$

```
SELECT ?nome ?valore
WHERE{
    ?nome rdf:type mine:Cryptocurrency .
    ?nome mine:Value ?valore
    FILTER(?valore > 25000)
}
```

nome	valore
Bitcoin	"34025.28"^^<http://www.w3.org/2001/XMLSchema#float>

Figura 9: Risultato della query.

2) Criptovalute basate su blockchain "classica"

```
SELECT ?nome ?tecnologia
WHERE{
    ?nome rdf:type mine:Cryptocurrency .
    ?nome mine:basedOnTechnology ?tecnologia .
    ?tecnologia rdf:type mine:Blockchain

}
```

nome	tecnologia
Bitcoin	BitcoinBlockchain
CARDANO	CardanoBlockchain

Figura 10: Risultato della query.

3) Per ogni moneta fornisce nome tecnologia, valore, gestore e data del prima vendita del token

```
SELECT ?nome ?tecnologia ?valore ?creatore ?token ?dataprimavendita
WHERE{
    ?nome rdf:type mine:Cryptocurrency .
    ?nome mine:Value ?valore.
    ?nome mine:basedOnTechnology ?tecnologia.
    ?nome mine:managedBy ?creatore .
    ?nome mine:hasToken ?token .
    ?token mine:startSellingDate ?dataprimavendita
}
```

nome	tecnologia	valore	gestore	token	
Monero	MoneroBlockchain	"180.0"^^<l	"Nobody"	XMRT	"2014-04-18T09:00:00"
Ethereum	Tangle	"2041.56"^	"Vitalik Buterin"^^	ETH	"2015-08-07T09:00:00"
CARDANO	CardanoBlockchain	"1.22"^^<ht	"CardanoGroup"	ADA	"2017-09-27T09:00:00"
Bitcoin	BitcoinBlockchain	"34025.28'	"Nobody"^^<http:	BTCT	"2009-01-12T09:00:00"

Figura 11: Risultato della query.

4) Trovare piattaforma di mining e problema matematico associato

```
SELECT ?nome ?tecnologia ?problema
WHERE{
  ?nome rdf:type mine:Cryptocurrency .
  OPTIONAL{?nome mine:miningBasedOnProblem ?problema.}
  OPTIONAL{?nome mine:minableWith ?tecnologia}
}
```

nome	tecnologia	problema
Monero	NVIDIAGpu	HashGuessing
Ethereum		
CARDANO		
Bitcoin	NVIDIAGpu	HashGuessing

Figura 12: Risultato della query, si può osservare la presenza di "Optional" che permette il ritorno di quelle valute che non hanno un problema matematico in quanto vi è associato un processo di tipo P.o.S.

5) Trovare le Criptovalute basate su architettura BitcoinCore

```
SELECT ?crypto
WHERE{
    ?crypto mine:basedOnTechnology ?tech.
    ?tech mine:hasArchitecture mine:BitcoinCore
}
```

crypto
Bitcoin

Figura 13: Risultato della query.

6) Trovare le Criptovalute con metodo di validazione Bitcoin Proof of Work

```
SELECT ?nome
    WHERE{
    ?nome rdf:type mine:Cryptocurrency .
    ?nome mine:basedOnTechnology ?tech .
    ?tech mine:hasValidationMethod mine:BitcoinProofOfWork
}
```

nome
Bitcoin

Figura 14: Risultato della query.

7) Trovare le Criptovalute con metodo di validazione Proof of Work (Generico)

```
SELECT ?nome
    WHERE{
    ?nome rdf:type mine:Cryptocurrency .
    ?nome mine:basedOnTechnology ?tech .
    ?tech mine:hasValidationMethod ?valid .
    ?valid rdf:type mine:ProofOfWork
}
```

nome
Bitcoin
Monero

Figura 15: Risultato della query.

8) Trovare le Criptovalute con metodo di validazione Proof of Stake (Generico)

```
SELECT ?nome
    WHERE{
    ?nome rdf:type mine:Cryptocurrency .
    ?nome mine:basedOnTechnology ?tech .
    ?tech mine:hasValidationMethod ?valid .
    ?valid rdf:type mine:ProofOfStake
}
```

nome
Ethereum
CARDANO

Figura 16: Risultato della query.

9) Trovare il linguaggio di scripting di una criptovaluta

```
SELECT ?nome ?script
WHERE{
    ?nome rdf:type mine:Cryptocurrency .
    ?nome mine:basedOnTechnology ?tecnologia .
    OPTIONAL {?tecnologia  mine:hasScriptingLanguage ?script}

}
```

nome	script
Bitcoin	Script
Ethereum	Solidity
Monero	
CARDANO	Plutus

Figura 17: Risultato della query, anche in questo caso si può osservare il ruolo di "Optional".

10) Monete che hanno data prima vendita diversa dalla data di creazione

```
SELECT ?nome  ?valore  ?datacreazione ?dataprimavendita
    WHERE{
    ?nome rdf:type mine:Cryptocurrency .
    ?nome mine:Value ?valore.
    ?nome mine:hasToken ?token.
    ?token mine:startSellingDate ?dataprimavendita.
    ?nome mine:creationDate ?datacreazione.
    FILTER(?dataprimavendita > ?datacreazione)
}
```

nome	valore	datacreazione	dataprimavendita
Ethereum	"2041.56"^	"2015-07-30T09:00:00"	"2015-08-07T09:00:00"
Bitcoin	"34025.28"	"2009-01-03T09:00:00"	"2009-01-12T09:00:00"

Figura 18: Risultato della query.

5 Conclusioni

In questo progetto si è analizzata la fattibilità e si è implementata un Ontologia per raccogliere la conoscenza sull'argomento delle criptovalute.

Il flusso di lavoro ha previsto la redazione di una narrazione, quindi di un diagramma ER e infine l'implementazione vera e propria che ha compreso la cura delle varie classi e proprietà includendo anche il collegamento con DBPedia.

Riferimenti bibliografici

[1] *bitcoin/bitcoin*. original-date: 2010-12-19T15:16:43Z. Giu. 2021. URL: `https://github.com/bitcoin/bitcoin` (visitato il 24/06/2021).

[2] *Script - Bitcoin*. URL: `https://en.bitcoin.it/wiki/Script` (visitato il 24/06/2021).