An Ontology of Bitcoin

Semantic Web Project Final

Zachary Rogers

08/21/2018

Johns Hopkins University

605.643

Blockchain technology is a fascinating yet confusing innovation. In the 10 years since Bitcoin was created, many new blockchain projects have launched. Some of these focus on the storing and transferring of value while others secure digital assets or provide private communication protocols. There are various consensus mechanisms these platforms use to secure and validate transactions on the network. Decentralized applications and smart contracts are new capabilities offered by more recent blockchain platforms such as Ethereum. In some cases, they have even been touted as the backbone of the new internet and the future of governance and corporate structure. With all this excitement about blockchain and the rapid innovation taking place, the technology has become a wild west of opportunity and fraud. The goal of this project is to focus on the basic concepts and components of blockchain and starts at the beginning, with Bitcoin. I chose Bitcoin because it is the original blockchain use-case. It first came into existence in 2008, when an anonymous person or group of people released a white paper online outlining a project called Bitcoin: ‘A Peer-to-Peer Electronic Cash System’ (6). Today it is the most well-known and valuable of all the crypto-currency and blockchain related projects. In 2017, the price of one Bitcoin rocketed from under $1,000 to nearly $20,000. Needless to say there has been a lot of new interest in the space.

The purpose of this project is to create a blockchain domain ontology specific to Bitcoin. My ontology will focus on the core components, abstracting away some of the more complex and technical details. I will not cover BTC currency production from bitcoin mining or the consensus mechanism that helps secure the network at this time. However I will be updating the model to include these and other details in the future. The main problem that is being solved here is to provide a better learning resource for beginners to the space to understand the key features of blockchain technology without being overwhelmed with complicated technical details. This is an important goal of this project.

There are two main types of components or entities that make up the core features of this Bitcoin Ontology; Classes and Properties. The Class entities that define these core features are Account, Block, Blockchain, Node, and Transaction. Each class has data and object properties that describe them. These entities can be combined into ‘Triples’ or subject-predicate-object expressions that encapsulates a statement about the entities and build up a vocabulary about the ontology topic (5).

Here is a breakdown of the classes and properties in my ontology:

Main Class Entity Definitions:

- Account - An account is identified and secured by a public and private key pair. Each account has a balance. Accounts can send and receive transactions.
- Block - A block is set of timestamped transactions. Each blocks is 2MB in size.
- Blockchain - A public ledger of chronologically timestamped blocks of transactions.
- Node - A Node processes blocks and adds them to the blockchain. A node is also an account.
- Transaction - A message sent from one account to another that can contain BTC tokens.

Class Hierarchy:

- Account
    - Balance
    - PublicKey
- Block
    - Hash
    - Header
    - CreationTime
    - Difficulty
    - Number
    - Size
- Blockchain
- Node
    - NodeAddress
- Transaction
    - Amount
    - Block_Header
    - Message
    - Receiving_Address

  - Sending_Address
  - Timestamp

Data Properties:

- TransactionDataProperty
  - transactionError
  - transactionStatus
  - transactionHash
  - transactionMessage
- AccountDataProperty
  - address
  - accountBalance
  - accountPublicKey
- NodeDataProperty
- BlockDataProperty
  - blockHash
  - blockHeader
  - blockCreationTime
  - blockSize
  - blockDifficulty
  - blockNumber

Object Properties:

- BlockchainObjectProperty
  - containsBlock
- AccountObjectProperty
  - sends
- BlockObjectProperty
- NodeObjectProperty
  - discoversBlock
  - minesFor
  - hasAccountAddress
- TransactionObjectProperty
  - to
  - from

Similar to the internet in its early days, the potential impact and use-cases for blockchain technology are not yet apparent. A large group of people are not yet aware of the technology nor of possible benefits and innovations it may bring them and society in the future. Before we can get there though, the knowledge base of the domain must evolve. Adoption of blockchain technology will come from user growth and applications first. Any helpful material and educational resources that can be provided will positively impact that progress.

While the blockchain space is new, there have been some prior efforts made to encapsulate the key concepts of blockchain technology into a semantic ontology. One such excellent work is called EthOn or Ethereum Ontology (4). This ontology model is specific to the Ethereum platform and goes into extreme detail with every single aspect of the Ethereum blockchain, consensus mechanism, mining, smart contracts and decentralized application development. It also covers topics such as Gas, which is a utility currency on the Ethereum network that must be spent to send a transaction which helps mitigate spamming and congestion on the network. I highly recommend taking a look at the EthOn, but it will be overwhelming for folks new to the space.

My ontology is leveraged from some of the work done with EthOn, but is focused specifically on the basics of the Bitcoin blockchain and the primary Classes and Properties that comprise it. The model is designed to outline the entities and properties of the architecture and will serve as a gentle introduction to understanding the key aspects of blockchain technology in a simple, understandable semantic ontology and visual graph. It is built using Protege version 5 with OWL syntax. With this gentle introduction to the key components of Bitcoin's blockchain, I hope to make the learning curve a little easier and inspire new interest.

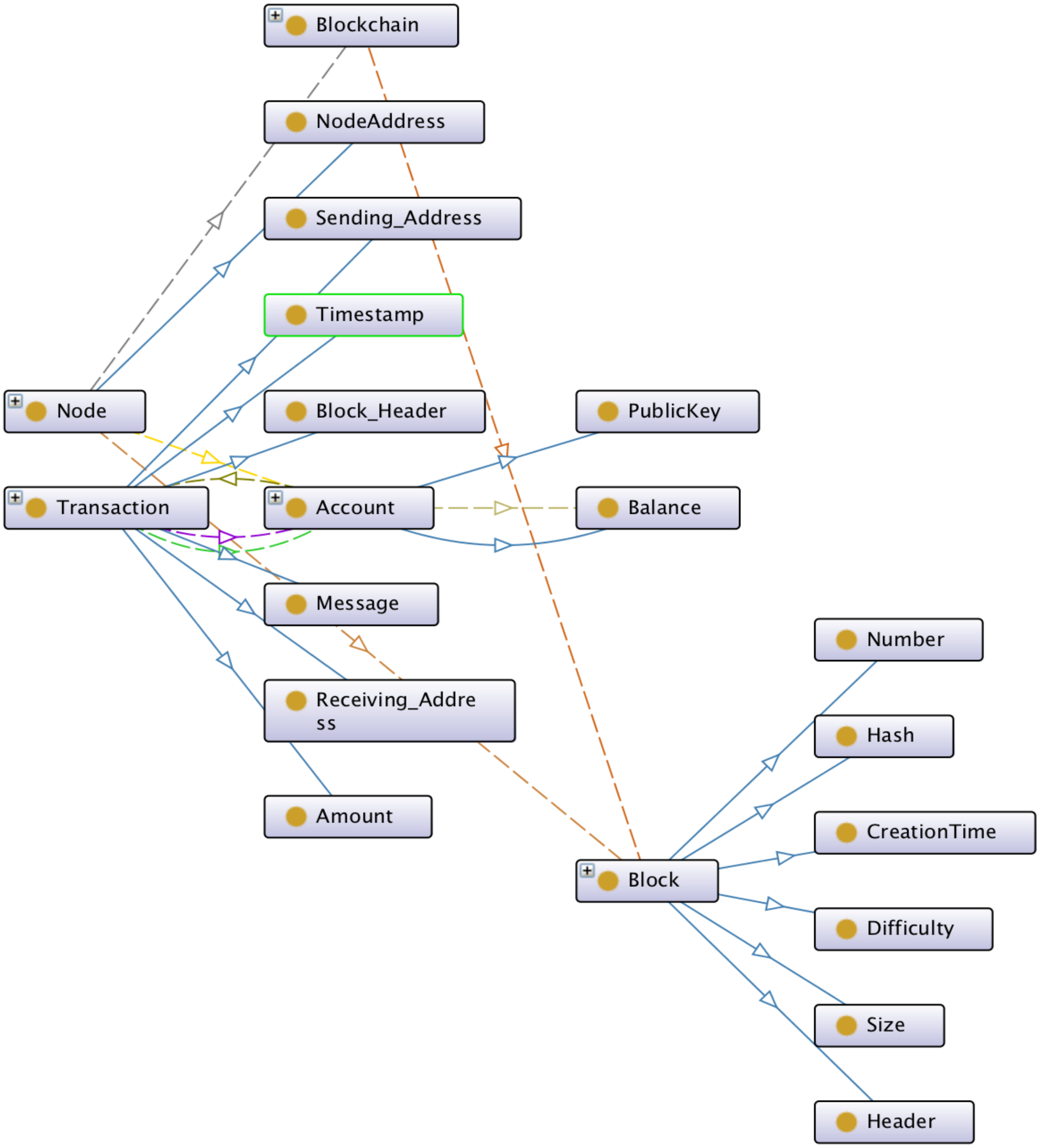

Figure 1: Graph Visualization of An Ontology of Bitcoin (Created with Protege OntoGraf plugin).

An ontology is a set of concepts in a domain or subject area that describes classes or concepts about the domain, and their properties and the relations between (9). An ontology together with a set of individual instances of classes constitutes a knowledge base (8). Using our ontology, we can make inferences about the domain that showcase it’s utility. Here are some example entity implementations and inferences using RDFS notation:

1) These are the classes of my blockchain platform ontology and their implementation.

```
@prefix :
<http://www.semanticweb.org/zrogers/ontologies/2018/7/blockchain-ontology/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
[ rdf:type owl:AllDisjointClasses ;
  owl:members ( :Account :Block :Blockchain :Message :Node ) ] .
```

2) In this example, the Account class checks it’s balance.

```
@prefix :
<http://www.semanticweb.org/zrogers/ontologies/2018/7/blockchain-ontology/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
:Account rdf:type owl:Class ;
        owl:equivalentClass [
                Rdf:type owl:Class ;
                owl:oneOf ( :address :checkBalance :Balance) ] .
```

3) This example implements a DataProperty, which is getting the balance of an account. The domain of this implementation is the class Account and the range is a Literal, which is a string or integer.

```
@prefix :
<http://www.semanticweb.org/zrogers/ontologies/2018/7/blockchain-ontology/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
:Balance rdf:type owl:DataProperty,
        owl:FunctionalProperty ;
        rdfs:domain :Account
        rdfs: range rdfs:literal .
```

4) This example shows a triple. An account sends a message. A message can be a text message or it can include a transaction amount.

```
@prefix :
<http://www.semanticweb.org/zrogers/ontologies/2018/7/blockchain-ontology/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
:sends a owl:ObjectProperty ;
        rdfs:domain :Account ;
        rdfs:range :Message ;
        rdfs:subPropertyOf :Transaction;
```

5) This is an example triple with two distinct classes. A node mines on a blockchain. This relationship is interesting in blockchains because a node could be mining on a different version of the blockchain software. If the versions are different enough and enough miners are mining it, this could cause a hard fork in the blockchain, where two different blockchains with the same history can exist in parallel.

```
@prefix :
<http://www.semanticweb.org/zrogers/ontologies/2018/7/blockchain-ontology/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
```

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
:minesFor rdf:type owl:ObjectProperty,
        rdfs:domain :Node
        rdfs:range :Blockchain .
```

SPARQL is a query language that can be used to write queries against data in a key-value format, such as RDF syntax like used in this ontology. I’ve written a couple of example queries to demonstrate the usage:

1) Select all the subclasses of a class in my ontology:

```
SELECT ?subclass
WHERE {
        ?subclass rdfs:subClassOf* :Transaction .
}
```

2) Select all addresses with a balance greater than zero, sorted by balance in descending order.

```
SELECT ?address ?balance
WHERE  {
        ?x Account:address ?address .
        ?x Account:balance ?balance .
        FILTER(?balance > 0) .
}
ORDER BY DESC(?balance)
```

3) Select all addresses that a given address “KEY” has sent transactions to. Includes the balance and transaction timestamp and orders results by timestamp in descending order. *Replace “KEY” below with the actual Public Key of the Account you want to query.

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?address ?receiving_address ?balance ?timestamp
WHERE {
        ?x Account:address ?address
        ?x Account:balance ?balance
        ?x Transaction:receiving_Address ?address
        ?x Transaction:timestamp ? timestamp
        FILTER( ?timestamp > “2018-01-01”^^xsd:dateTime && ?address =
        “KEY”^^xsd:string))
    }
    ORDER BY DESC(?timestamp)
```

In conclusion, I believe blockchain technology will continue to play a larger and more important role in the near future. Although the cryptocurrency space has been fraught with speculation and irrational ambitions, I do truly believe in the long term value of this revolutionary technology and want to understand as much about it as is possible. I have also gone through the learning curve to understand the basic aspects of blockchain tech and know how confusing the space can be at first. I hope this project will serve as a useful learning resource for students and those interested in the technology.

Ultimately, this model will act as a high level starting point for many people new to the space. I plan to further develop and add features to my ontology and I am working on developing an interactive webpage to host it. Users will be able to click into each entity in the ontology and see it’s sub and super classes, detailed descriptions of each entity with usage examples of its properties including domain and rage, as well as a formal implementation in RDFS and OWL syntaxes. The final webpage will also display an interactive graph UI of the ontology with the same underlying information. I am also thinking of ways to open-source my ontology. It would be great if anybody with knowledge of the space can help develop and improve upon the model. Keep an eye out online for An Ontology of Bitcoin coming soon to a web browser near you.

Sources:

(1) "XML Schema Datatypes in RDF and OWL." *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, W3C, www.w3.org/TR/swbp-xsch-datatypes/.

(2) "Protege4Pizzas10Minutes." *Protege Ontology Library - Protege Wiki*, protegewiki.stanford.edu/wiki/Protege4Pizzas10Minutes.

(3) "DOACC in OWL." *DOACC Project*, doacc.github.io/concepts/doacc-owl.html.

(4) *EthOn: Ethereum Ontology*, ethon.consensys.net/entities-az.html.

(5) "Semantic Triple." *Wikipedia*, Wikimedia Foundation, 27 July 2018, en.wikipedia.org/wiki/Semantic_triple.

(6) Nakamoto, Satohi. "Bitcoin: A Peer-to-Peer Electronic Cash System." October, 31 2008, https://bitcoin.org/bitcoin.pdf

(7) "Ontology Development 101: A Guide to Creating Your First Ontology." *Protégé*, protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html.

(8) "Ontology." *Wikipedia*, Wikimedia Foundation, 18 Aug. 2018, en.wikipedia.org/wiki/Ontology.