Implementation of REA Contracts as Blockchain Smart Contracts: An Exploratory Example

By

Graham Gal
University of Massachusetts

and

William E. McCarthy
Michigan State University

January 2018

Introduction

With the advent of Bitcoin and its blockchain trading platform, other implementations have been considered. For instance, other cryptocurrencies have been created as groups to recognize their ability to raise large amounts of capital outside of traditional markets such as the NYSE, FTSE, NASDAQ, etc. One consistent feature of these cryptocurrencies is that the markets for their exchanges are somewhat unregulated. Specifically, holders of these cryptocurrencies have very little recourse over the performance of activities by the issuers. That is, there does not exist a mechanism to enforce reciprocal actions by the issuers. This situation has given rise to an approach to create contracts for the performance of actions. While contracts are not a new construct, their inclusion as part of a blockchain implementation certainly are. The term "Smart Contract" has been coined to indicate a creation of contract which is coded and included as part of a blockchain.

Smart Contracts are a group of conditions which have been embedded in computer code, added as a transaction to a blockchain, and mined into a block on the chain. This approach has a number of benefits. First, by coding all the conditions of the contract and then compiling this code into an executable program, it is not possible to alter the terms of the contract without changing the compiled code. Second, by adding this code to the blockchain, all participants or nodes on the blockchain have a record of the transaction or contract, and therefore parties to the contract are not able to change any of the terms without changing the information contained on all the nodes of the blockchain. Third, the mining of the block makes the transactions in the block immutable. While this is similar to the benefits of the transactions being held by all nodes, the mining of blocks ensure the unalterable nature of all transactions. A final benefit of the creation and the addition of the contract to a blockchain is that actions taken by parties to fulfill their commitments under the contract, and added to the blockchain are also immutable and can also be universally understood to either meet or not meet the conditions enumerated in the contract. This universal and automatic determination that a transaction is consistent with the conditions of the contract is precisely what makes the coded contract "Smart". While the creation and addition of the coded Smart Contracts to a blockchain solves certain concerns of the exchange of assets, there are other issues that need to be resolved.

When Bitcoin is exchanged between wallets on its blockchain there is little disagreement that one party relinquishes a certain amount of the cryptocurrency and another wallet now has control of those bitcoins. Bitcoins only exist on its blockchain. When other assets are to be exchanged they do not "exist" on the blockchain, only records would exist; although there is some question about where records would actually exist. Another question concerns the suitability and precise structure of the assets to be exchanged. For instance, a Smart-Contract for a discreet asset such as an egg might seem to be quite easy to code. However, there is a question of whether any instance of an egg fulfills this commitment. This can become far more complex when other assets are to be exchanged in the Smart Contract. If a car is to be exchanged the asset is far more complex as it is an aggregation of many components, each of which can impacts whether or not the car actually exchanged meets the commitment as enumerated in the Smart Contract. While off blockchain contracts have existed for quite some time, there are also a plethora of enforcement environments. For these Smart Contracts to function there needs to be a single agreed upon structure that can be coded exactly. The REA Ontology can provide such a mechanism.

Figure 1 (chapter 5 figure 8) depicts the three layers of the REA ontology. The middle or scheduling layer can also be viewed as the contracting layer. Contracts can be considered as a special case of a schedule as a contract presents a template or expectation of future events. A schedule or contract is composed of a bundle of commitments which will be fulfilled by a series of economic events. Based on the terms of the contract, a particular event will actually fulfill a commitment. For instance, some contract terms will be fulfilled when the merchandise leaves the warehouse while in other cases the fulfillment occurs when the merchandise is received. From the perspective of a Smart Contract, the events which fulfill the specific commitment must be recorded on the blockchain and connected to the commitment clauses in the Smart Contract. The coding of the particular type of economic event must be included in the coding of the contract which must then determine whether the specific event is an instance of the type of event specified. For assets such as a car there can be a more detailed aggregations of components which could be part of the bundle of commitments in the Smart Contract, so the particulars of a certain contract can get quite entailed. The parties to contracts can get as detailed as necessary, and the REA ontology can accommodate the various commitments which will be bundled in the contract.

Our Exploratory Example

In Figure 2, we present an exploratory REA contract example that we intend to discuss at the VMBO workshop. It is an M1 model of a simple contracted exchange of merchandise for currency with an additional late-delivery penalty[1]. Because of space limitations, the number of classes, associations, and attributes have been deliberately minimized in this UML diagram, so they fit on a single page. This minimization is especially prominent on the "unhappy path" (i.e., the delivery penalty) whose associations are outlined with red lines. In Figure 1, we illustrated the independent-view REA metamodel (M2); here we move down a level to an M1 example. It is our intent to construct some actual smart-contract components for the Figure 2 example, using the Ethereum language *Solidity* (Nash 2017).

Our discussion at the Amsterdam VMBO, will encompass a number of issues, both theoretical and practical, as we map an REA Contract to one or more smart contracts. Among these discussion issues will be the following:

1. **How do you identify instances of the REA categories in collaboration space?** The attributes shown in Figure 2 are stubs that need more research and development analysis, especially with regard to linking blockchain objects (instances) to each other.
2. **What is the proper mixture of resources and resource-types in a blockchain transaction?** In trading-partner REA usage, it is often the case that some economic resources (like commodities, liquids, gases, and currency) are type-specified at the scheduling layer and then type-flowed at the accountability level. Our example illustrates both cases. Is this how blockchain contracts and transactions are supposed to work?
3. **Given the limited capacity at present of blockchain transactions at present, do we expect many policy layer and even scheduling layer REA components (yellow and red in the figures) to be enumerated off-chain for the near future and how will all of these be linked?** This question may very well have different answers in the case of permissioned vs. non-permissioned blockchains.

---

[1] This example is derived from the automobile industry via AIAG (the Automobile Industry Action Group).

4. **Do most smart contracts work on the notion that custody of the cryptocurrency is transferred initially to the contract as the trusted third party and then an occurrence signal from the opposite economic event triggers a release of this currency?** This seems to be a key and necessary component of the "trust" factor for anonymous collaboration space to work well.

5. **Do we need a separate instance of a smart contract for each set of paired commitments in an REA contract?:** Figure 2 portrays an exchange on the happy path where the merchandise shipment occurs as scheduled, which in turn leaves the contracted payment penalty unfulfilled. Do multiple commitment pairs necessitate multiple smart contracts and how are their triggered or dormant operations linked to each other?

6. **What are the representation issues and linking issues involved in "omission" economic events and their commitment images (like the "potential failure to ship on time" class shown at the bottom left of Figure 2)?** Actually, this is an area that needs analysis for trading partner REA models as well, but its difficulties seem harder in collaboration space.

As we state above, we hope to have many of these questions partially analyzed at the time of the workshop, and we hope to use the expertise of VMBO attendees to analyze all of them.

REFERENCES:

Nash, Gerald, "Build Your First Smart Contract" , December 2017, https://medium.com/crypto-currently/build-your-first-smart-contract-fc36a8ff50ca

McCarthy, William E, Guido Geerts, and Graham Gal (2017) *The REA Accounting Model as an Accounting and Economic Ontology,* monograph manuscript submitted for publication to the American Accounting Association and Working Paper, Michigan State University, East Lansing, MI, USA, July 2017.