

Bridging the Gap between Denotational Semantics and Operational Semantics in Smart Contract

Wong Wai Wai*

Multimedia University, Faculty of Law, Malaysia

Gan Chin Lay

Multimedia University, Faculty of Business, Malaysia

Abstract

In this era of increasing cyber dependency in business dealings there is huge potential in the adoption of Distributed Ledger Technologies (DLT) particularly in the context of smart contract in the commercial world. The phenomenon of smart contract operates independently without the cumbersome need to engage any intermediary. It has been argued that there is too much dependency on the programming aspect in the creation of smart contracts by programmers and computer scientists. Smart contract are more like an Apps which is capable of executing specific task but it fails to observe the fundamental understanding and agreement between the negotiating parties which is the core essence in traditional commercial contract. The objectives of this paper are to first, demonstrate the semantic discrepancies between traditional contract and smart contracts and the implication of the latter. Secondly, to support the proposition that programmers and computer scientists lack the required legal knowledge and logic in appreciating the various legal terms and effects of a concluded contract, there is a need to include lawyers and regulators to enhance the drafting of the corresponding denotational semantics in programming smart contract. This paper contends that the operational semantics which deals with the execution of the contract on technical platform should be consistent with and correspond with the denotational semantics.

Keywords: Smart contract; Denotational semantics; Operational semantics.



CC BY: [Creative Commons Attribution License 4.0s](https://creativecommons.org/licenses/by/4.0/)

1. Introduction

Technology based on blockchain and Distributed Ledger Technologies are capable of being adopted by several industries and sectors in achieving its numerous applications. The application of such largely involve sectors such as banking, finance, health, network connectivity, public sector, security and authentication systems. Smart contracts are the very core of this potential innovation of DLT (Khalil *et al.*, 2017). However, if smart contract is to be regarded as a form of legal contract and is enforceable through computer program and code, its operational semantics which deals with the execution on its technical platform should be consistent with and correspond with the denotational semantics. This paper is divided into four parts. First, this paper will briefly explain blockchain and DLT. Secondly, it will examine the difference between legal contract and smart contract. Thirdly, this paper will highlight the translation problem in creating and designing smart contracts by programmers and computer scientists. Fourthly, the discrepancies are illustrated by the comparative analysis between traditional legal contractual principles and smart contract code through real smart contract applications. Finally, this paper conclude with its recommendations to the translation problem in order to bridge the gap between denotational semantics and operational semantics in smart contracts.

2. Blockchain and Distributed Ledger Technology (“DLT”)

The concept of blockchain was introduced some ten years ago in 2008 by Satoshi Nakamoto, the creator of the first digital currency (bitcoin) (Nakamoto, 2008). Blockchain is a digitized, decentralized, public ledger growing on blocks which are recorded and added to it in chronological order. DLT is the general term that is used to refer to methods of maintaining distributed ledgers on networks of computers. The term ‘blockchain’ is often used interchangeably with DLT. Such usage is not accurate because blockchain represents only one type of DLT. Many of the best-known DLTs are based on a blockchain approach. But there are other types of DLT that are not built on blockchain technology. The difference between blockchain and DLT are that DLT is board term used to describe technologies that distribute records or information amongst those who are using it, whether privately or publicly. Blockchain is just a sub- category of this board term DLT (Thake, 2018).

The below definition gives a more comprehensive explanation as to what blockchain is:

A blockchain is a globally shared, transactional database. This means that everyone can read the entries in the database just by participating in the network. If you want to change something in the database, you need to create a so-called transaction which is required to be accepted by all others. The word “transaction” implies that the change you want to make (assume you want to change two values at the same time) is either not done at all or completely

applied. Furthermore, while your transaction is applied to the database, no other transaction can alter it ([Introduction to Smart Contracts — Solidity 0.5.0 documentation, \(n.d.\)](#)).

DLT is basically a digital system for recording and storing information of various transactions. The records include the particulars and details of the transactions undertaken. DLT records and store the transaction details in different places at the same time. As oppose to traditional centralized databases, distributed ledgers have no central data storage or administration functionality. The perceived benefits of DLT for information sharing amongst various institutions are not well understood, and may prevent the adoption of this technology that can enable institutions to share vital digital information ([Ølnes et al., 2017](#)). Sectors ranging from banking, real estate, medical insurance and health care can reap the benefits of using emerging technologies such as blockchain and smart contracts to increase efficiencies, eradicate potential erroneous data, and eliminating issues relating to data integrity ([Kuo et al., 2017](#)). DLT also has the potential for a wide range of commercial and economic transactions in areas such as governance, institutional and collaborative economics ([Davidson et al., 2016](#)). For instance, one of the most well-known application of blockchain, i.e. cryptocurrencies such as Bitcoin, if used as a financial medium can transform the banking industry in many ways, from enabling peer-to-peer encrypted monetary transactions and asset transfers in a highly securely platform at minimal fees ([Collomb, 2016](#)); ([Team, 2018](#)).

Despite the transformative potential of DLT and blockchain, various adoption challenges exist. Uncertainties pertaining to the reliability and maturity of the technology, risk associated with the changes to standard industry practices, insufficient understanding and clarify of the positive impact and business gains, and most importantly, legal ramifications of using these emerging technologies are some of the crucial areas that need to be addressed by all parties involved prior to adoption and acceptance ([Deshpande et al., 2017](#)). Despite these uncertainties, DLT and blockchain technologies are here to stay, largely in part due to their ability to enable global transactions that are secure, fast, transparent, and trustworthy.

3. Legal Contract and Smart Contract

The development and implementation of DLT and Smart Contract in the commercial sector require us to rethink seriously the way we carry out such business transaction and our perception idea of business relationship. There are a lot of “floating” issues surrounding the adoption of smart contract. For instance, what really is a smart contract; what are the legal considerations of such smart contract, how may smart contract interact with legal contract and the ultimate question is whether smart contract can eventually replace the existing universally well accepted legal contract. To answer these questions, we need to start with the very basic understanding of what legal contract and smart contract is.

Legal contract is defined as:

A contract is an agreement giving rise to obligations which are enforced or recognised by law. The factor which distinguishes contractual from other legal obligations is that they are based on the agreement of the contracting parties ([Treitel, 2003](#)).

On the other hand, a smart contract is defined as:

“Smart contract is an automatable and enforceable agreement. Automatable by computer, although some parts may require human input and control. Enforceable either by legal enforcement of rights and obligations or via tamper-proof execution of computer code.” ([Clack et al., 2016](#)).

The implementation of smart contract is through Solidity. Solidity is a type of programming language which was designed to allow smart contract code to be executed on the Ethereum platform. Ethereum is a decentralized platform for applications to be executed exactly as programmed without any chance of fraud and third-party interference. **Ethereum** is described as:

“... that runs smart contracts: applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third party interference. These apps run on a custom built blockchain, an enormously powerful shared global infrastructure that can over value around and represent the ownership of property. This enables developers to create markets, store registries of debts or promises, move funds in accordance with instructions given long in the past (like a will or a futures contract) and many other things that have not been invented yet, all without a middle man or counterparty risk.” ([Ethereum Project, n.d.](#))

Base on the above definitions, legal contract and smart contract are different but they overlap with one another. Confusion further arises when people have different conceptions as to what smart contract is. The computer scientists often use the term smart contract in a different way as compare with lawyers and regulators. For lawyers, the term contract connotes and provides evidence of a very particular legal relationship between the parties to the transaction. But for computer scientists, they often think of smart contract in the context of code. Stark (2016) presents two distinct schools of thought to view smart contracts as:

“(1) **Smart legal contract**: The term smart contract is used to refer to legal contract or elements of legal contracts, being represented and executed by software.

(2) **Smart contract code**: The other (*school of thought*) relates it more to a piece of code (known as a software agent) that is designed to execute certain tasks if pre-defined conditions are met. Such tasks are often embedded within, and performed on a distributed ledger.” ([Stark, 2016](#)).

When smart contract is understood as smart contract code, it is no more than a set of codes which are written to execute certain functions. It bears no legal agreement or relationship between the parties of the smart contract. Smart contract is merely an **execution mechanism** for a set of deterministic obligations, rather than the contract itself ([Lim et al., 2016](#)). The smart contract includes a sequence of instructions updating and executing the normative states

(obligation, prohibition and permissions). However, the order of the sequence of instructions in smart contract does not reflect accurately the logical order of the contractual sequence expressed in natural and legal language.

The controversy and challenge arises if and when smart contract is to be understood as smart legal contract. It is because for it to be treated as legal contract, it should have the elements of contracts in it. These elements are being electronically executed. For instance, in the law of contract, the requirements that need to be satisfied for a valid contract inter-alia are offer, acceptance, consideration and intention to create legal relationship. This is how lawyers and regulators think, interpret and believe smart legal contract to be so. Lawyers and regulators believe that no matter whether a transaction takes place in the real physical world or in the electronic world, all the legal requirements are needed to be satisfied in order to make it into a binding and enforceable contract.

However, for a smart legal contract to be implemented, it is only require to be attached to one or more pieces of code designed to execute certain tasks. And when the pre-defined conditions are met, smart legal contracts are therefore functionally comprised of various pieces of smart contract code (Whitepaper, 2017). Furthermore, on top of that, it creates legally enforceable rights too. And therefore, Stark's two distinct schools of thought to view smart contract is used merely for the purpose of reflecting and explaining the different usage of the term 'smart contract'. The two different schools of thought did not accurately define what smart contract is. If we were to fall back on the definition given by Clack *et al.* (2016) that "Smart contract is an automatable and enforceable agreement it meant that smart contract is automatable by computer, although some parts may require human input and control and it is enforceable either by legal enforcement of rights and obligations or via tamper-proof execution of computer code", it actually describes smart contract as a combination of enforceable legal agreement and computer code. Hence in reality, there is an inseparable relationship between legally enforceable agreement and computer code in smart contract.

4. Comparative Analysis

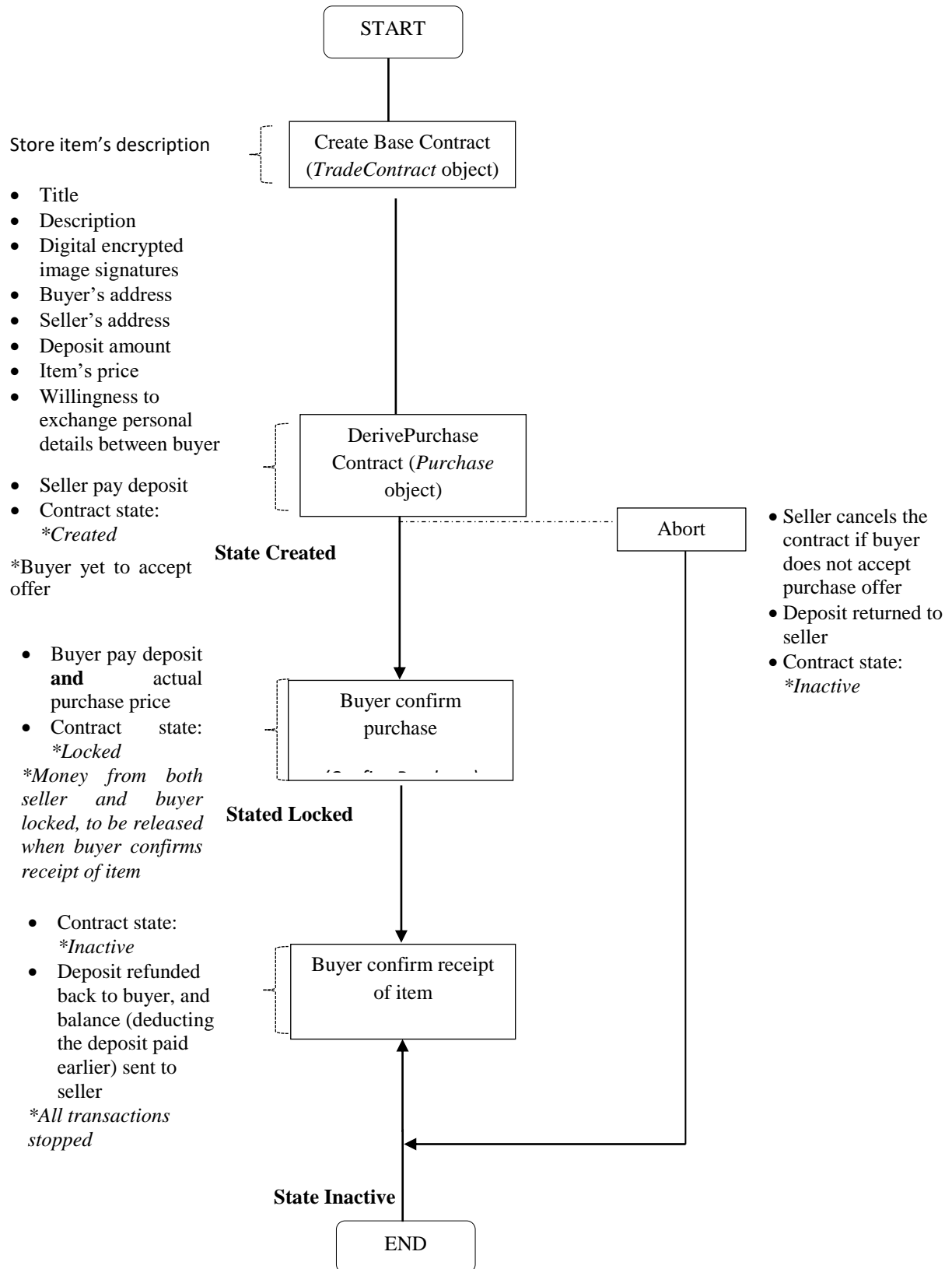
The following comparative analysis between a smart contract and traditional contract is to explore and to demonstrate the semantic discrepancies between smart legal contract and traditional legal contract principles. A simple sample of a purchase contract is adopted from Ethereum.io ("Solidity — Solidity 0.4.25 documentation," n.d.) for the purpose of comparison.

The Purchase Contract

```

1  Contract Purchase is TradeContract {
2  Function Purchase (string title, string description, bool verify, bytes32[] imageSignatures) payable
3  Trade Contract ( title, description , verify , msg . value / 2 , msg . value / 2 , imageSignatures ) {
4  if ( 2 * price != msg . value ) throw ; }
5  event purchaseConfirmed ( ) ;
6  event itemReceived ( ) ;
7  function abort ( )
8  only Seller
9  in State ( State . Created ) {
10 aborted ( ) ;
11 state = State . Inactive ;
12 if ( ! seller . send ( this . balance ) ) throw ; }
13 function confirmPurchase ( )
14 in State ( State . Created )
15 require ( msg . value == 2 * price )
16 payable {
17 purchaseConfirmed ( ) ;
18 buyer = msg . sender ;
19 state = State . Locked ; }
20 function confirmReceived ( )
21 only Buyer
22 in State ( State . Locked ) {
23 itemReceived ( ) ;
24 state = State . Inactive ;
25 if ( ! buyer . send ( deposit ) || ! seller . send ( this . balance ) ) throw ; } }
```

Below is the purchase contract represents in a diagram form for easy reference.



From the above illustration we can see that the purchase contract starts when users setting up the 'Trade Contract' with the relevant information such as title, description, digital encrypted image signatures, buyer's address, seller's address, deposit amount and item's price. In a purchase smart contract, a safe purchase is ensured by requiring the seller as well as the buyer to pay a deposit before the intended transaction take place. The subject matter of the trade contract such as the item or goods will only be released upon the buyers 'Confirmation Receipt' of the item or goods. Seller is required to pay a deposit, acting as an incentive not to betray a buyer (Niya et al., 2017). The seller has to transfer Ether in the contractor of the contract. The value transferred will be divided into two. When the buyer executes the 'Confirm Purchase' function, the buyer needs to pay a deposit in addition to the actual purchase price. When it is done, the status of the contract is 'State Locked' Locking of the deposits is to make

sure that no party will break their promise and betray the other side since both parties have deposited the same amount of money in the smart contract and neither of them will get their deposit back until the task has been fulfilled (Niya *et al.*, 2017). When the contract is still at the 'State Created', it means the buyer has not accepted the offer and seller can execute 'abort' function and the deposit stored in the contract is refunded to the seller then the status will become 'State Inactive' (Niya *et al.*, 2017). On the other hand, if the transaction proceeds and in the 'Confirm Received' function, the buyer confirms that he/she has received the item, the deposit is then to be refunded to the buyer and the rest of the balance stored in the contract is sent to the seller. The status of the smart contract is set to be 'State Inactive' which means no further interaction (Niya *et al.*, 2017).

The above has illustrated the process of a simple purchase smart contract in Ethereum. As we can see, the smart contract is similar to the traditional legal contract in many ways. In the basic principles of English contract law that we all are familiar with, the formation of the contract includes these basic elements such as offer, acceptance, consideration, and contractual intention. Some of these elements can be seen in smart contract. Offer made by the seller with all the relevant information of the item or goods and the seller create a base contract. That is similar to the offer that we traditionally understood from the English Law. However, under the English law, an offer must be distinguished from an invitation to treat in which a person does not make an offer but merely invites another party who is interested to make an offer as stated in the case of *Carlill v Carbolic Smoke Ball Company* [1893] 2 QB 256. Therefore, a seller does not necessarily make an offer. An interested buyer will make an offer to the seller and the seller accepts the offer under the traditional scenario. In a smart purchase contract as illustrated above, there is also an offer and acceptance. However, the offer is usually made by the seller and acceptance by the buyer differs from some of the traditional scenarios. Furthermore, offer can be terminated if it has not been accepted by the other party. An offer may be revoked at any time before acceptance and an offer is made irrevocable by acceptance. These principles are illustrated in the case of *Offord v Davies* (1873) 1 J.R. 73 (N.Z.) These principles are equally adopted in smart contract. In smart contract, the seller can make an offer. He can revoke or cancel the offer when the buyer does not accept the purchase offer. However, once the buyer accepts the contract by paying deposit and the actual purchase price, the smart contract is locked and no one is able to withdraw or cancel the contract.

Under the traditional English law, consideration is important for contract because without a consideration, the contract would not be binding. Consideration is something that is of value which gives rise to a promise to be enforceable. Consideration is applicable in smart contract. The party, namely the buyer must make deposit together with the purchase price deposited in smart contract (Kim and Laskowski, n.d.) for the purpose of executing the smart contract transaction.

Another important element under the traditional English law is the intention to create legal relationship. Intention to create legal relationship creates a legally binding agreement or contract. Furthermore, it also indicates the readiness of a party to accept the legal consequence of having entered into an agreement. However, the element of intention to create legal relationship between the parties is not clear in smart contract. The question is whether it is necessary for the parties to the smart contract to have the intention to create legal relationship because smart contract is an automated execution of certain task. Once the task has been executed, the terms of the smart contract are performed. There is no issue of non performance or partial performance of the smart contract. The smart contract is designed in a way that the deposit required by both parties ensures performance and compliance of the smart contract.

The law of contract under the English law provides redress for numerous issues arising from a concluded contract. For example, fraudulent misrepresentation occurs when one party to a contract knowingly makes an untrue statement of fact or presentation which induces the other party to enter into that contract. The implication of a fraudulent misrepresentation in contract is that the injured party can claim compensation for all losses resulting from the fraudulent misrepresentation. Could fraudulent misrepresentation occur in smart contract? The answer to this is yes because in smart contract, the seller discloses information, description and digital images of the item or goods in question. A Seller with bad faith will intentionally disclose inaccurate or fraudulent information, description and digital images of the item or goods to be sold in order to induce the other party to enter into a smart contract. Although the smart contract allows the Purchaser to 'Confirm Received' the item or goods before advancing to the next stage of 'State Inactive', nevertheless, if the Purchaser was to subsequently after paying for the item or goods, discover that the item or the goods are not what he/she has contracted to buy as envisaged, the Purchaser in a Smart Contract has no way to seek remedy from the Seller in the smart contract. The smart contract itself provides no further mechanism or process to deal with any legal issues arising pursuant to the smart contract. The implication of this is that smart contract lacks the flexibility that is inherent to the existing law and legal principles governing traditional contract.

It can be concluded from the above comparative analysis between smart contract and traditional legal contract which reaffirmed the proposition of this paper that smart contract possibly view as smart legal contract because it is similar to a legal contract or that smart contract contains some elements of legal contracts being represented and executed by software and computer codes. Based on the above example of a simple purchase contract, smart contract contains the elements of offer, acceptance, termination of an offer and consideration. However, smart contracts do not contain the element of intention to create legal relationship and it provides the parties with no remedy or redress. The issue is whether it is possible to make a smart contract more of a legal contract and therefore render it legally enforceable under the traditional contractual principles. In other words, can smart contract become a real contract? There are many factors contributing to this but one of the factors that would be highlighted in this paper is the translation problem. Translation problem in the present context means the difficulty to translate the relevant laws and legal principle of contract accurately into the programming of smart contract.

5. Translation Problem

Take for example if the seller of goods initiates a smart contract with a purchaser, the payment is executed in codes and transacted automatically upon delivery. What if the purchaser would like to include an indemnity clause? What if the purchaser after receiving the goods for some time, later discovers that the goods are defective? Can the purchaser insist to include an indemnity or be able to sue the seller? The answers to these questions are unlikely because first, clauses are not part of the smart contract code and they are not something that can be self-executed (Gillioz, 2017). Secondly, smart contract provides no remedy or dispute resolution mechanism whenever dispute arises between the parties. When designing the smart contract programme, there was too much dependency on the programming aspect in creating smart contracts by programmers and computer scientists (Khalil *et al.*, 2017). Much attention has been centered on the programming aspect of creating smart contract. But programmers and computer scientists lack the required level of legal knowledge, training and logic in appreciating the various legal terms and principles behind a legally enforceable contract, smart contract that is created solely by programmers and computer scientists may run into the risk of lacking denotational semantic that corresponds to the legal semantics (Khalil *et al.*, 2017).

If one accepts the contention that smart contract is not merely a set of computer code but a smart legal contract which contains obligations and legal terms that are enforceable, hence, in programming or writing smart contract, one must ensure that the software developers who design smart contract take note of the legal rules and principles behind the specific type of contract in question. What we are trying to argue is that there is a gap between a smart contract's legal semantics, which is a methodology of legal comparison, jurisprudence, legal translation, regulatory semantics with denotational semantics. In essence, if smart contract is to be regarded as smart legal contract or a binding enforceable contract, it must be expressed in a specific programming language that constitutes the substance of the legal norms. Hence, in order for smart contracts to fulfil their role in this context, it must have precise algorithms or computer program code that represent the law and legal principles. Without the precise algorithms or computer code corresponding to the legal semantics, it will provide room for potential abuses and exploitation of loopholes. Therefore, the accuracy of the code is very important in order to ensure the legal certainty that the program in smart contract to be executed according to the intention of the parties and to comply with the terms and conditions of the contract (Peters and Panayi, 2016).

It is important to take cognizance of the limitations of smart contract and to understand that there are many complicated contractual relationship that are not suitable for performance through smart contract. Smart contracts should only be used for simple and straight forward transactions. Commercial agreements that contain many clauses which protect parties from various liabilities are not suitable for execution through code because not all clauses in contracts are susceptible to automation and self-execution (Subassandran, 2018).

One should also take note of the different objectives between smart contract and legal contract. Smart contract is to ensure compliance and all actions are automated in the digital environment whereas legal contract is a product of legal principles established in the physical world and it serve as a fundamentally remedial institution to resolve dispute between the parties (Werbach and Cornell, 2017).

6. Recommendations

Progressing from the above discussions in relation to the translation problem of smart contract, this paper offers few possible solutions to this. The operational semantics which deals with the execution of the contract on technical platform should be consistent and ought to correspond with the denotational semantics. In order to achieve that, this paper proposes that there is a need to include lawyers and regulators to enhance the drafting of the corresponding denotational semantics in programming smart contract. Lawyers and regulators should be engaged in the early stages when designing smart contract such as identifying the type of agreements, possible legal implications, alternative and remedial actions of an agreement.

To illustrate this, using the earlier example of the Seller initiating a smart contract with the Purchaser for the supply of goods. To make the smart contract more into a legal contract, an indemnity clause may be inserted at the initiation stage at the "*create base contract*" by the seller. The Clause may read as follows: that after physically receiving the goods, the customer has the right to check the nature, characteristics and the functioning of the goods and seek the indemnity from the seller if the goods are not of merchantable quality after payment has been made and received. This is to provide some remedial measures after the Purchaser have paid for the goods and later discover that the goods are defective. Indemnity clause can become part of the terms or description of the offer given by the seller. Alternatively, smart contract should stipulate a grace period after purchaser's '*Confirm Received*' and '*State Inactive*' stages. This is to allow redress if there is any dispute between the parties, for the smart contract is still active. One may even design or program the smart contract in such as way that to demand more deposit from the seller in order to ensure merchantable quality of the goods offered for sale.

The smart contract should also support customer withdrawals after a period time stipulated by the seller after payment and physical goods have been exchanged. In some circumstances, 'Oracle' has been used to confirm certain status in relation to the smart contract (Singh, 2018). However, it is almost impossible to implement oracle to automatically check conditions and provides input to the Smart Contract in cases where good received are defective (Ruiz, 2017). When the item or goods received are found to be defective, it still requires actual communication or dealing between the parties. This can be performed by way of follow-on contract. **Follow-on contract** is part of smart contract that programmed to allow humans to take part in it. The issue as to whether follow-on contracts are legally binding is a complex question to be answered (Giancaspro, 2018).

The above suggestions such as the insertion of an indemnity clause, stipulate grace period and support withdrawals in smart contracts are derived from the traditional contract laws and legal principles. The point we want to emphasize here is that programmers and computer scientists are not aware of the various legal terms and principles behind a legally enforceable contract. Programmers and computer scientists would not be able to design a smart contract that consists of indemnity clause, stipulate grace period and support withdrawals. They will only be able to do so after receiving proper advice from lawyers and regulators. When lawyers and regulators participate in the designing and drafting of smart contract, lawyers and regulators input is to be exchanged with the programmers and computer scientists. This exchange is to ensure that the operational semantics which concern with the execution of the contract on technical platform to be consistent and correspond with the denotational semantics that is legal meaning of the contract. That is to incorporate the laws and legal principles as far as possible into smart contract.

In a complicated or complex transaction which may involve the use of numerous clauses, it is suggested that parties may want to consider the use of the text-versions of agreements on top of the smart contract so that they can read and agree on the terms, memorialize terms as smart contracts are not equipped to address the desired contract with all the terms. It is always better to have a piece of document that can be enforceable in the courts of law (Levi and Lipton, 2018). In this scenario, smart contract is treated more like a set of smart contract code which executes certain functions such as to ensure payment and delivery. Smart contract in this context merely provide a appliance for the automatic execution of a contract (Whitepaper, 2017). If there is any discrepancy between the code executed and the legal contract, the legal contract ought to prevail. For example, if for some reason the legal contract does not match with the smart contract code, the legal contract version should prevail no matter what the Solidity code says (or executes), it is invalid in case it does not match with the natural language version of legal contract (Ruiz, 2017).

7. Conclusion

Smart contracts has introduced into the commercial world a position where smart contract code is capable of becoming a binding contract. This transformation is phenomenon and significant particularly in the context of contract law. Algorithms of smart contract need to be precise in order to avoid abuses and loopholes. As programmers and computer scientists lack the required legal knowledge, there is a need to include lawyers and regulators to enhance the designing of the corresponding denotational semantics in programming smart contract. On the other hand, lawyers and regulators should have the initiative to learn about the operational semantics of smart contract in order to appreciate its functionality. When smart contract have no consistency and correspondence with the law and legal principles, smart contract can only be regarded as set of computer code execute functions that has been pre-determined by the programmers or computer scientists. It can never be regarded as a binding contract in the legal sense. Although there is a lot of uncertainty and complexity surrounding smart contract and its development, it is important to take note of the limitations and nature of smart contract. Smart contract and Distributed Ledger Technologies (DLT) are in their development stage and there is a great potential for this technology to have a positive impact in the modern cyber and technology based society.

References

- Clack, C. D., Bakshi, V. A. and Braine, L. (2016). Smart Contract Templates: foundations, design landscape and research directions.: Available: <http://arxiv.org/abs/1608.00771>
- Collomb, A. (2016). Blockchain / Distributed Ledger Technology (DLT): What Impact on the Financial Sector? *Digiworld Economic Journal*: Available: www.comstrat.org
- Davidson, S., De Filippi, P. and Potts, J. (2016). Disrupting governance: The new institutional economics of distributed ledger technology. *SSRN Electronic Journal*: Available: <https://doi.org/10.2139/ssrn.2811995>
- Deshpande, A., Stewart, K., Lepetit, L. and Gunashekar, S. (2017). Distributed Ledger Technologies/Blockchain: Challenges, opportunities and the prospects for standards. Available: https://www.bsigroup.com/LocalFiles/zh-tw/InfoSec-newsletter/No201706/download/BSI_Blockchain_DLT_Web.pdf
- Ethereum Project (n.d.). Available: <https://www.ethereum.org/>
- Giancaspro, M. (2018). That contract your computer made could get you in a legal bind.: Available: <https://phys.org/news/2018-04-legal.html>
- Gillioz, F. (2017). From Smart Contract to Smart Legal Contract. Available: <https://www.linkedin.com/pulse/from-smart-contract-legal-fabien-gillioz>
- Introduction to Smart Contracts — Solidity 0.5.0 documentation ((n.d.)). Available: <https://solidity.readthedocs.io/en/develop/introduction-to-smart-contracts.html>
- Khalil, F. A., Ceci, M., O'Brien, L. and Butler, T. (2017). A solution for the problems of translation and transparency in smart contracts. *Grctc.Com*: Available: <http://www.grctc.com/wp-content/uploads/2017/06/GRCTC-Smart-Contracts-White-Paper-2017.pdf>
- Kim, H. and Laskowski, M. (n.d.). A Perspective on Blockchain Smart Contracts. *Reducing Uncertainty and Complexity in Value Exchange*.: Available: <https://arxiv.org/ftp/arxiv/papers/1801/1801.02029.pdf>
- Kuo, T.-T., Kim, H.-E. and Ohno-Machado, L. (2017). Blockchain distributed ledger technologies for biomedical and health care applications. *Journal of the American Medical Informatics Association*, 24(6): 1211–20. Available: <https://doi.org/10.1093/jamia/ocx068>

- Levi, S. D. and Lipton, A. B. (2018). An Introduction to Smart Contracts and Their Potential and Inherent Limitations.: Available: <https://corpgov.law.harvard.edu/2018/05/26/an-introduction-to-smart-contracts-and-their-potential-and-inherent-limitations/>
- Lim, C., Saw, T. and Sargeant, C. (2016). Smart contracts: Bridging the gap between expectation and reality oxford law faculty. Available: <https://www.law.ox.ac.uk/business-law-blog/blog/2016/07/smart-contracts-bridging-gap-between-expectation-and-reality>
- Nakamoto, S. (2008). Bitcoin, A peer-to-peer electronic cash system. Available: http://www.academia.edu/download/32413652/BitCoin_P2P_electronic_cash_system.pdf
- Niya, S., Schüpfer, F., Bocek, T. and Stiller, B. (2017). A peer-to-peer purchase and rental smart contract-based application. Files.Ifz.Uzh.Ch. . Available: <https://files.ifi.uzh.ch/CSG/staff/Rafati/Purchase-Rental-APP-SC.pdf>
- Ølnes, S., Ubacht, J. and Janssen, M. (2017). Blockchain in government, Benefits and implications of distributed ledger technology for information sharing. *Government Information Quarterly*, 34(3): 355–64. Available: <https://doi.org/10.1016/J.GIQ.2017.09.007>
- Peters, G. W. and Panayi, E. (2016). Understanding modern banking ledgers through blockchain technologies. *Future of Transaction Processing and Smart Contracts on the Internet of Money*: 239–78. Available: https://doi.org/10.1007/978-3-319-42448-4_13
- Ruiz, J. (2017). Smart contracts for business-to-consumer trading, Not legal contracts and not very smart. Available: <https://www.linkedin.com/pulse/smart-contracts-business-to-consumer-trading-legal-very-jesus-ruiz>
- Singh, A. (2018). What is an oracle? How oracles connect smart contracts to the real world? | blockchain semantics blog. Available: <https://www.blockchainsemantics.com/blog/oracle-connects-smart-contracts-to-real-world/>
- Stark, J. (2016). Making sense of blockchain smart contracts - coindesk,. Available: <https://www.coindesk.com/making-sense-smart-contracts/>
- Subassandran, R. (2018). Blockchain and smart contracts, Today and tomorrow blockchain. Available: <http://www.coindesk.com/making-sense-smart->
- Team, F. (2018). Top 9 advantages of cryptocurrency as a financial medium,. Available: <https://blog.finjan.com/advantages-of-cryptocurrency/>
- Thake, M. (2018). What's the difference between blockchain and DLT? – nakamo.to – Medium. Available: <https://medium.com/nakamo-to/whats-the-difference-between-blockchain-and-dlt-e4b9312c75dd>
- Treitel, G. H. (2003). The law of contract. *Thomson/Sweet & Maxwell*: Available: https://www.google.com/search?q=The+Law+of+Contract%2C+11%22+ed%2C+treitel&rlz=1C1GCEU_enMY819MY819&oq=The+Law+of+Contract%2C+11%22+ed%2C+treitel&aqs=chrome..69i57j69i60j0.783j0j4&sourceid=chrome&ie=UTF-8
- Werbach, K. and Cornell, N. (2017). Contracts Ex Machina. *Duke Law Journal*, 67: 313-82. Available: <https://heinonline.org/HOL/Page?handle=hein.journals/duk1r67&id=321&div=11&collection=journals>
- Whitepaper (2017). Smart Contracts and Distributed Ledger-A Legal Perspective. Available: <http://www2.isda.org/attachment/ODcwMA==/>