Research and Development of Intelligent Contract Auditing Platform Based on Artificial Intelligence Technology

Yuqing Zhang[1,a], Jia Wang[1,b], Hui Zeng[1,c], Fangyuan Wu[1,d], Pinxiao Guan[2,e*]

[a]3240457088@qq.com , [b]2052535650@qq.com , [c]422801513@qq.com, [d]1191436481@qq.com , [e]1320508450@qq.com.

[1]Hebei University of Economics and Trade, Shijiazhuang, China, Faculty of Law
[2]University of Melbourne, Melbourne, Australia, Law School

Abstract. The contract is an important legal element directly related to the interests of small and medium-sized enterprises (SMEs) and individuals. In the new era, we use artificial intelligence technology to solve contract problems in practice and realize the purpose of legal aid. The intelligent contract review platform utilizes several high-tech, based on the actual existence of the relevant issues to achieve the purpose of the research and development to reduce the risk of the contract, and at the same time to show the system's mode of operation, operational efficiency.

Keywords: Contract review; Risk control; Artificial intelligence; Legal assistance; SMEs

1 Introduction

In today's age of artificial intelligence, the authenticity of contracts and the prevention of audit risks are crucial to the smooth fulfillment of transactions.[1]However, it is difficult for non-professionals to see the potential risks in the contract. The intelligent contract audit platform covers the electronic management of the whole life cycle of the contract from contract drafting, auditing, signing, fulfillment, and archiving, which can successfully solve many problems such as low efficiency of manual audit and low-risk warning ability in contract management and provide solid and reliable legal risk prevention and control guarantee.

MSIEID 2023, December 08-10, Guangzhou, People's Republic of China

DOI 10.4108/eai.8-12-2023.2344809

2 Contract Audit Platform Implementation Path Planning

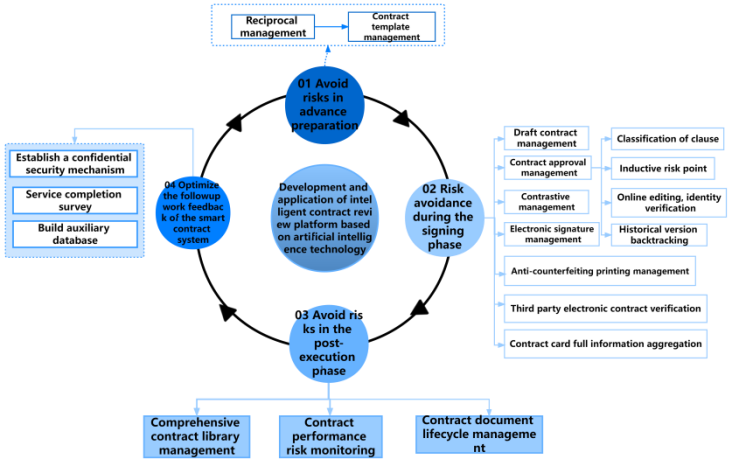

Figure 1:Functional frameworks

The Figure 1 above is the basic functional framework of the smart contract review platform. This paper will start with the core functions of artificial intelligence to introduce the feasibility of the realization of smart contracts.

2.1 State Machines

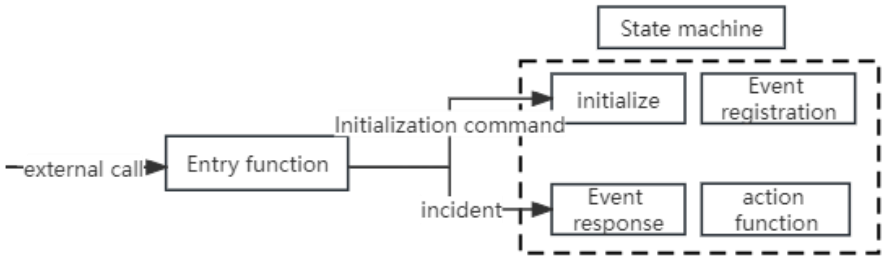

Figure 2: State machine program architecture

The platform is based on chaincode's state machine, which enables state transfer by storing the state of smart contracts on the blockchain.[2] As shown in Figure 2, the entry function is called externally. The state machine includes an initialization module, event response module, event registration module, and action function.

- The initialization module is responsible for constructing a contract instance and all contracts;
- The event registration module is responsible for responding to changes, i.e. by establishing a state transition matrix;

- The event response module modifies the state of the state machine according to the state transition matrix and input events;[3]
- The action function implements specific actions, such as buy_ticket, deposit, and so on.

Taking airline delay insurance as an example, smart contracts can judge the probability of compensation by identifying aviation information. In a smart contract system, flight information is written to a state machine and blockchain, and this data is used to determine whether a flight is delayed, increasing the credibility of the results.

2.2 Blockchain

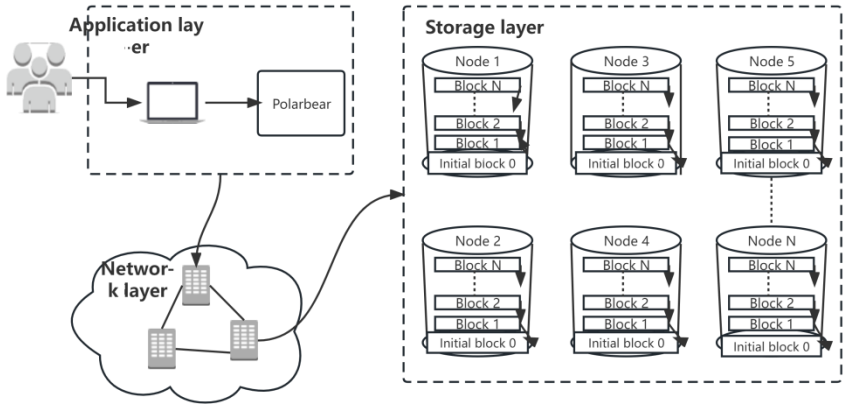

Figure 3: Blockchain architecture

As shown in Figure 3, the platform uses the decentralized, trustless and immutable characteristics of the blockchain network to set a firewall between the system server and other computers to make security policies.

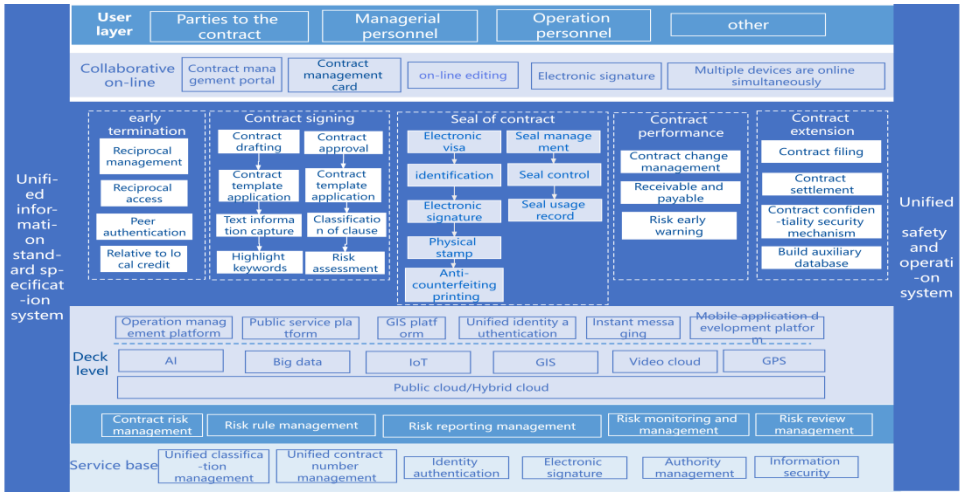

Figure 4: Quasi-implementation architecture

From a technical point of view, smart contracts, as shown in Figure 4, include five elements: data layer, transport layer, smart contract body, verification layer, and execution layer.

The data layer is the data storage place of the smart contract, and the code is deployed in the ledger in a distributed and replicable manner, responsible for the functions of receiving, processing, storing, and sending, to achieve the automated execution of the contract terms.[4] Each node in the blockchain network records transaction information for each part, through a specific hash algorithm, these points are connected and form a block structure, from the perspective of cryptography theory, forming a decentralized database, information can be truly symmetric and shared, at the same time, each node synchronously updates the transaction information, reducing the possibility of tampering. Avoid mistakes. So the contract code stored there, shuttled between each blockchain node, is read recognized, and executed by the computer. Functionally similar to a paper contract text, stylization carries the rights and obligations of the parties to the contract.

The transport layer can be thought of as a static database of smart contracts, which collects all the rules for calling, executing, and communicating smart contracts. In short, the content of the contract is determined by the parties through negotiation, and only the rights and obligations of the parties concerned need to be negotiated based on legal rules, industry norms, etc. On this basis, professional technicians are asked to put the contract in the form of a digital code

The content is written in the blockchain in the form of "if-then" computer language, enabling the transformation of natural language descriptions and contract code construction.[5]

The main body of a smart contract includes two parts: agreement and parameter. Agreement is a procedural legal text description issued by a standards body, that is, a formatted contract template; Parameters include business logic module (main parameter) and various auxiliary modules, such as data management module, user management module, and contract management module. The business logic module is essentially a digital contract text based on the negotiation between the two parties. The attachment module is to supplement and improve the smart contract based on user needs and application scenarios. For example, the data management module can realize data management and cleaning; The user management module can authenticate the management identity and authority of the contract subject. It is also a participant in itself, and can both store and respond to the information it receives.

The verification layer is mainly composed of some verification algorithms to ensure the consistency between the generated code of the contract and the contract text, and its function is to avoid the disclosure of the privacy of the transaction subject and ensure the security of the transaction process.[6] The blockchain uses an asymmetric encryption algorithm to obtain a pair of public and private keys for identification. Among them, the public key publishes the account address on the blockchain network; A private key is a user's password to access an account, private information that can only be known by the user. For example, if A pairs a transaction message with B's public key with his private key, every node in the blockchain receives this message, but it can only be decrypted if the private key is enabled by B.

The execution layer is the running layer of the smart contract. Only when there are no defects in the current four processes, the smart contract will enter the execution stage. Compared with the traditional contract, its self-execution characteristics are more prominent. In other words,

the execution of smart contracts is automatic, once the conditions are met, it will start to execute, and cannot be undone, it is this irrevocable self-execution, that effectively improves the efficiency of contract performance, eliminates human interference, and promotes the establishment of decentralized distributed trust system.

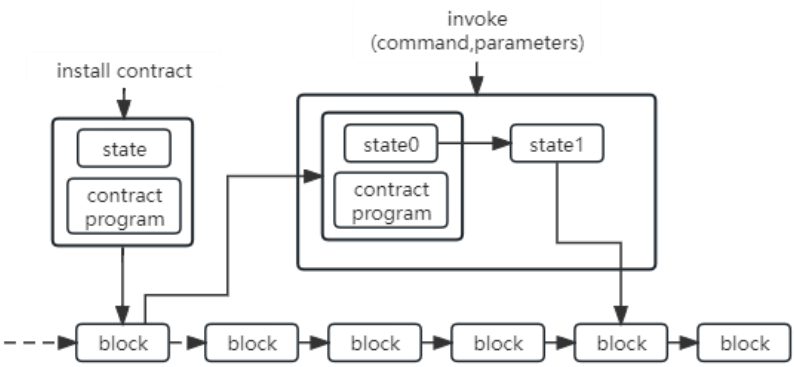

Figure 5: The process of executing smart contracts on blockchain platforms

As shown in Figure 5, the contract published on the chain has an initial state, and each call (including instructions and parameters) needs to read the contract logic and the previous state. After execution, the new state is stored in the block, and the smart contract is the state machine running on the blockchain.

2.3 General Algorithm

A smart contract is a state machine and therefore necessarily contains a state machine and a set of interface functions. Every call to a smart contract goes through an interface function, which is equivalent to applying an action to a state machine. The state machine will fulfill the promises contained in the contract, as well as the constraints between the promises. The following code describes the function call process of a smart contract.

Algorithm 1: Manage function calls with the other party

```
// Define a function to query the reputation and operation of the legal partner based on the enterprise name or social credit code.
function queryPartnerInfo(name or code) {
    var result = {};
    // Call the SOA interface to interact with the various enterprise search platforms
    var data = call SOA(name or code); if (data.isValid); }
    if (data.isValid()) {
        // Parse the data to extract information on subject qualification, capacity, authorization, etc.
        result.qualification = parseQualification(data);
```

```
        result.capability = parseCapability(data); result.
        result.authorization = parseAuthorization(data); result.
        authorization = parseAuthorization(data); result.
    } else {
        throw new Exception("Invalid input or query failed"); }
    }
}
// Define a function to display the results of the query.
function display result(result) {
    if (result.isEmpty()) {
        if (result.isEmpty()) { alert("No information found"); } else {
    } else {
        show(result.qualification);
        show(result.capability); } else { show(result.authorization); show(result.capability); show(result.authorization)
        show(result.authorization); }
        }
}
// Define a function to handle user input and output.
function handleUserInput() {
    // Get the user's business name or social credit code entered in the search box.
    var nameOrCode = getInput(); // Call the query function to get and retrieve the result.
    // Call the query function to get and retrieve the results of the query
    var result = queryPartnerInfo(name or code); // Call the query function and get the result.
    display result(result);
}
```

Algorithm 2: Contract performance risk monitoring

```
# Define a contract risk monitoring function and pass in a contract object.
function monitor_contract_risk(contract):
// Integrate with financial system to obtain key terms information such as contract expiration, receipt and payment, invoice, delivery and default
key_terms = get_key_terms_from_finance_system(contract)
```

// Obtain credit information and capital status from external data sources

counterparty_info = get_counterparty_info_from_external_source(contract)

// Set the buried points and combine the key points of the contract with the contract progress to form a list

checkpoints = combine_key_points_and_progress(key_terms)

// Walk through each buried point to determine whether the threshold is exceeded

for each checkpoint in checkpoints:

// If the value exceeds the threshold, for example, the contract is not performed after the expiration, and the key nodes such as delivery, receipt and payment have passed

if checkpoint exceeds threshold:

// Assess the risk level according to the contract terms and the information of the other party

risk_level = evaluate_risk_level(checkpoint, key_terms, counterparty_info)

// Generate warning and reminder content according to the risk level

alert_content = generate_alert_content(risk_level, checkpoint)

// According to the contract object, obtain the contact information of the relevant personnel

contact_info = get_contact_info_from_contract(contract)

// Send early warning reminders to relevant personnel

send_alert_content(alert_content, contact_info)

return result

}

2.4 Evaluation of contract results

Based on the above state machine and blockchain capabilities, if the performance of a smart contract is evaluated, three metrics will be used: Precision: P, Recall: R, and F1 value. The calculation formula is as follows:

$$P=\frac{TP}{TP+FP}\times 100\%$$

$$R=\frac{TP}{TP+FN}\times 100\%$$

$$F1=\frac{2*P*R}{P+R}\times 100\%$$

TP represents the number of samples in the contract text where the entity is actually labeled positive and the model predicts that the label is also positive, TN represents the number of samples in the contract text where the entity is actually labeled negative and the model predicts that the label is also negative. FP represents the entity in the contract text that is

actually labeled negative, but the model predicts the number of samples that are labeled positive; FN indicates that the entity in the contract text is actually labeled positive, but the model predicts the number of samples labeled negative.[7]

3 Prospects and Challenges

This platform can help small and micro enterprises and the general public to prevent contract risks by identifying them at a lower cost for legal assistance through the use of artificial intelligence in specific legal affairs. For the legal practice community, a more efficient and stable contract review model can be explored to reduce repetitive and mechanical legal document review work. Enterprises can effectively reduce legal costs and contract risks.For academics, it is a new attempt to combine the disciplines of law and computer science, exploring the new path of "artificial intelligence + law" and helping to promote the intersection of disciplines.

However, the risks of this platform should not be ignored. The artificial intelligence technology cited by the platform is prone to legal risks, and the challenge of artificial intelligence on data security cannot be ignored. If we do not have strong technical conditions and acquire personnel and technology through commercial forces, the risk of data leakage and misappropriation will continue to exist.For example, the hacker used the BEC token smart contract vulnerability to launch an attack on the BEC token, through the calculation of the overflow, out of thin air to generate 5.7896 × 1058 tokens, at the same time, the hacker will be these tokens transferred to the trading platform for sale, resulting in the market value of the BEC token almost to zero.The legal database is an important foundation for this platform to be able to operate properly, the more data we have, the more resources for the AI to learn, and the more the AI can carry out in-depth algorithmic learning, but it is not easy to collect a large amount of data, and the problem of lack of data still exists.

4 Conclusion

This platform is the theoretical knowledge in the field of contract, legal provisions, practical operation, etc. into one system, can use algorithms to comprehensively and extensively analyze all types of contracts, in the contract signing, performance and other stages can provide operational advice, provides a feasible risk prevention measures, thus maximizing the avoidance of contract risk, to facilitate the contract at the same time, reflecting the maintenance of the legitimate rights and interests of their party protection. Artificial intelligence technology as another wave of scientific and technological revolution of the theoretical achievements, in the current direction of the times, can and will play a greater role in the field of social sciences, this platform is the use of the intelligent era of science and technology disk, in the contract review, risk prediction, and so on many aspects of the provision of appropriate services to achieve the purpose of legal aid.

References

[1] WANG Puwei, YANG Aerospace, MENG Ji et al. Formal definition and reference implementation of contract-oriented smart contract[J]. Journal of Software,2019,30(09):2608-2619

[2] Liu Yuhang. Design and realization of contract order management system based on workflow high concurrency[D]. Beijing Jiaotong University,2019.

[3] M. Zhou, N. Duan, S. Liu, et al. Recent advances in neural natural language processing - modeling, training, and pushing[J].Engineering,2020,6(03):155-188.

[4] Naiyu Wang,Yuxin Ye,Lu Liu et al. Advances in language modeling based on deep learning[J]. Journal of Software,2021,32(04):1082-1115.

[5] Wright, Mark L. J. (Cited 15 times). The Pari Passu Clause in Sovereign Bond Contracts: Evolution or Intelligent Design [article]. The Three and a Half Minute Transaction: Boilerplate and the Limits of Contract Design, Hofstra Law Review, Vol. 40, Issue 1 (Fall 2011), pp. 103-114.

[6] Feliu, Silvia. Intelligent Agents and Consumer Protection [article]. Special Issue: Electronic Agents. International Journal of Law and Information Technology, Vol. 9, Issue 3 (2001), pp. 235-248.

[7] Banteka, Nadia (Cited 16 times). Artificially Intelligent Persons [article]. Houston Law Review, Vol. 58, Issue 3 (Winter 2021), pp. 537-596.