工程科学学报
Chinese Journal of Engineering

基于要约承诺的智能法律合约订立方法与实现

郭倩 朱岩 殷红建 陈娥 王迪 刘国伟

Design and implementation of conclusion procedure in smart legal contracts based on negotiation and acceptance

GUO Qian, ZHU Yan, YIN Hong-jian, CHEN E, WANG Di, LIU Guo-wei

引用本文:
郭倩, 朱岩, 殷红建, 陈娥, 王迪, 刘国伟. 基于要约－承诺的智能法律合约订立方法与实现[J]. 工程科学学报, 2022, 44(12): 2138-2153. doi: 10.13374/j.issn2095-9389.2021.04.08.005
GUO Qian, ZHU Yan, YIN Hong-jian, CHEN E, WANG Di, LIU Guo-wei. Design and implementation of conclusion procedure in smart legal contracts based on negotiation and acceptance[J]. *Chinese Journal of Engineering*, 2022, 44(12): 2138-2153. doi: 10.13374/j.issn2095-9389.2021.04.08.005

在线阅读 View online: https://doi.org/10.13374/j.issn2095-9389.2021.04.08.005

工程科学学报, 第 44 卷, 第 12 期: 2138−2153, 2022 年 12 月
Chinese Journal of Engineering, Vol. 44, No. 12: 2138−2153, December 2022
https://doi.org/10.13374/j.issn2095-9389.2021.04.08.005; http://cje.ustb.edu.cn

基于要约–承诺的智能法律合约订立方法与实现

郭　倩[1)], 朱　岩[1)✉], 殷红建[1)], 陈　娥[1)], 王　迪[1)], 刘国伟[2)]

1) 北京科技大学计算机与通信工程学院, 北京 100083　2) 北京市经济和信息化局, 北京 100744
✉通信作者, E-mail: zhuyan@ustb.edu.cn

摘　要　从合同订立的相关法律规定入手, 通过引入合约范本化思想, 提出了一种包含智能合约建立、部署、订立和存证四个阶段的规范化合约订立流程, 使之满足书面合同成立要件的法律规定; 同时, 在合约范本中提出了书面化交互接口, 使之满足合约“订”和“立”两个阶段的交互; 此外, 在智能法律合约语言 SPESC 中引入了合约订立相关语法, 使之满足合约订立过程中的“要约–承诺”制度, 并设计了三种区块链交易结构支持当事人注册、签名、条款执行中交互数据的存证; 最后, 以销售合约为实例, 从订立过程的要约认定、承诺认定、存证合法性三方面辨析了所提智能法律合约订立方案的合规性. 所做工作将有助于为智能法律合约的订立过程提供法律依据, 促进我国智能合约的法律化建设.

关键词　智能法律合约; 合同订立; 合约范本; 要约–承诺; 合规性; 法律存证
分类号　TP319

Design and implementation of conclusion procedure in smart legal contracts based on negotiation and acceptance

GUO Qian[1)], *ZHU Yan*[1)✉], *YIN Hong-jian*[1)], *CHEN E*[1)], *WANG Di*[1)], *LIU Guo-wei*[2)]

1) School of Computer & Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
2) Beijing Municipal Bureau of Economy and Information Technology, Beijing 100744, China
✉ Corresponding author, E-mail: zhuyan@ustb.edu.cn

ABSTRACT　Smart legal contract (SLC), as a form of smart contract in accordance with the law, has attracted extensive attention in recent years. However, the conclusion procedure of an SLC still lacks effective technical methods to make it conform to the current legal regulations, which directly affects the legitimacy of the SLC contract. Therefore, this paper starts with the relevant legal regulations of contract conclusion and introduces the idea of contract normative pattern (CNP), which is a reusable form, model, or template for contract conclusion. Based on these regulations, we propose a standardized conclusion procedure of smart legal contracts. This procedure includes four stages: establishment, deployment, conclusion, and deposit to meet the current legal regulations for the conditions of establishment in the written contracts. Meanwhile, a written form of interactive interface is proposed to satisfy the two stages of “negotiation” and “acceptance” in the CNP. The negotiation stage supports the parties to repeatedly negotiate and determine the pending contents in the CNP, and the acceptance stage is to activate the behavioral attribute by actively triggering predefined algorithms, such as registration and signature. In addition, the syntax of contract conclusion, including parties’ negotiation and acceptance, is introduced in the SLC language, SPESC, to adhere to the “negotiation-acceptance” mechanism in the conclusion procedure. In this paper, we design three blockchain’s transaction structures to store interactive data during party registration, signature, and clause execution. Finally, considering the sales contract as an example, we analyze the legitimacy of the proposed conclusion scheme based on three aspects: negotiation confirmation, acceptance confirmation, and deposit legitimacy. This paper will provide a legal basis for the

收稿日期: 2021−04−08
基金项目: 国家科技部重点研发计划资助项目(2018YFB1402702); 国家自然科学基金资助项目(61972032)

conclusion procedure of an SLC and promote better achievement of legalization of a smart contract.

KEY WORDS　smart legal contract; contract conclusion; contract normative pattern; negotiation–acceptance mechanism; legitimacy; deposit legitimacy

随着数字经济时代的到来，以区块链为基础的智能合约（Smart contract）[1−2]正成为构建“价值互联网”的颠覆性技术．伴随区块链技术[3]的不断演化，智能合约不仅是区块链上满足预定条件时自动执行的计算机代码，而且也演化出支持智能合约可执行程序开发、生成、部署、运行、验证的信息系统[4]，这使得区块链应用开发日益完善、产业应用日益广泛．

尽管智能合约具有将法律合同以程序代码形式加以自动执行的能力[5]，但就智能合约本身而言，它仍然采用常规计算机语言编写，与传统的程序代码并无差异，因此在可读性、易理解、法律效力等方面仍有别于法律合同．据此，智能法律合约（Smart legal contract）[6]被提出，它是一种含有合同构成要素、涵盖合同缔约方依据要约和承诺达成履行约定的计算机程序，兼具法律合同和计算机程序的特征，为代码法律化提供了基础．

智能法律合约依据法律规定在形式上能够以程序代码表达法律合同条款[7]，保证了它既具有现实合同的法律特征和易理解性，又有计算机程序的规范性，有助于解决智能合约的合法合规问题．按照我国现行法律规定，法律合同的订立过程也应遵守特定的法律原则，特别是订立过程的“要约–承诺”制度．然而，目前在学术和实践中都缺乏以计算机程序为对象的订立过程合规性研究，无法通过技术手段使智能法律合约的订立过程符合现行法律规定，进而保证合约合法生效．

针对上述智能法律合约订立过程中缺乏技术手段来保证其合规性的问题，本文从合同订立的相关法律规定入手，规范了智能合约订立流程，扩充了智能法律合约语言使之满足合约订立过程中的“要约–承诺”制度，并设计三种区块链交易结构支持订立过程中交互数据的存证．具体工作如下：

（1）通过引入合约范本化思想，提出了一种规范化的智能合约订立流程，该流程包含智能合约的建立、部署、订立和存证四个阶段，并详细给出智能合约订立的数据流程，保证了智能合约满足我国现行法律对书面合同成立条件的规定；

（2）在合约范本中提出了针对要素属性信息和行为属性信息两种类型的书面化交互接口，用于满足合约“订”和“立”两个阶段的数据交互，并通过销售合约实例给出了当事人注册与签名动作的行为处理算法，验证了上述交互接口设计实施合约订立过程的有效性；

（3）在智能法律合约语言 SPESC 中引入了合约订立相关语法，该语法包含当事人宣称和当事人签名两部分，用于缔约双方以意思表示的形式对要约和承诺过程中的事项进行表述，并可记录合约成立的当事人信息、签名和签名时间，保证智能法律合约能以书面合同样式体现“要约–承诺”制度．

本文以销售合同为实例对上述智能合约订立方法予以验证．首先，给出了采用 SPESC 语言所撰写的销售合约范本，并按照所提出的智能合约订立流程，实现了该合约从协商、注册、签名到执行的全流程；其次，设计了三种伴随交易结构用于当事人注册、签名、条款执行中所有交互数据的区块链存证；最后，以上述实例为基础，从订立过程的要约认定、承诺认定、存证合法性三方面辨析了所提出的智能法律合约订立方案的合规性．

1　相关工作

近几年智能合约法律化问题得到了广泛关注，不少学者从智能合约本身是否符合现行法律法规以及如何对智能合约进行规范使其转化为现行法律法规所认可的形式两个方面进行了智能合约法律化研究．

一方面，一些学者采用“直觉逻辑”研究智能合约本身所具有的合同属性，例如，2018 年 Kasprzyk 等[8]通过分析智能合约能否真实表达缔约双方的意图，来界定智能合约的法律效力．2019 年郭少飞[9]从合同效力，修改与履行，违约及救济三方面深入剖析智能合约的合同法适用性．同年，陈吉栋[10]通过讨论智能合约是否具有法律合同的要约–承诺构造来判断智能合约能否成为法律合同．上述研究基本认定了智能合约的合同法适用性．

另一方面，一些学者则采用“构造逻辑”研究智能合约法律化问题，希望通过对现有智能合约技术予以改进或规范化，使其转化为现行法律法规所认可的形式．现有研究大致可分为如下三个

方面:

首先, 从规范程序设计与平台构建角度, 为解决非计算机人员难以理解智能合约内容的问题, 高级智能合约语言被提出, 它是介于自然语言与智能合约语言间的一种语言. 2016 年 Farmer 和 Hu[11] 提出了一种具有精确语义的形式语言 FCL, 通过该语言编写的智能法律合约由一组包含定义、协议和规则的组件构成. 2018 年 He 等[12] 提出了一种智能合约规范化语言 SPESC, 它可以将现实合同采用类自然语言的形式编写为智能法律合约. 同年, Regnath 和 Steinhorst [13] 提出了 SmaCoNat 语言, 创建了从自然语言到程序语义的映射.

其次, 为使智能法律合约自动转化成与其意思表达一致的智能合约代码, 2017 年 Mavridou 和 Laszka [14] 提出了一种 FSolidM 语义框架, 用于将高级智能合约设计为有限状态机 FSM 模型, 使其自动生成以太坊 Solidity 合约. 2018 年 Choudhury 等[15] 提出了一种根据特定领域的本体和语义规则自动生成智能合约代码的框架; 2020 年 Zupan 等[16] 提出了一种基于 Petri 网生成智能合约的框架; 同年, Zhu 等[17] 提出了一种将高级智能合约语言 SPESC 自动转化为智能合约语言 Solidity 的转化规则.

最后, 从合约模板生成智能合约代码角度, 为使智能法律合约具备与现实合同同等的法律效力, 2016 至 2018 年间, Clack 等[18–20] 通过探索智能合约的语义框架, 并基于现实合同设计了具备法律效力的合约模板, 同时使用操作参数建立了高级智能合约与智能合约间的联系. Account 和 OpenLaw 项目开发了一种使用特殊标记语言的合约模板库, 将现实合同转化为对应的智能法律合约[21].

上述研究表明智能合约正朝着跨领域合作、标准统一、法律化的方向不断发展.

2　预备知识

解释 1: 合同订立

法律上, 法律合同订立是指缔约当事人相互为意思表示并达成合意而成立了合同. 合同的订立是合同双方动态行为和静态协议的统一, 它既包括缔约各方在达成协议之前接触和洽谈的整个动态的过程, 也包括双方达成合意、确定合同的主要条款或者合同的条款之后所形成的协议[22]. 也就是说, 合同订立分为“订”和“立”两个阶段在, 前者强调缔约双方在达成合意之前不断接触、协商的整个动态过程, 包括要约, 要约邀请等; 后者强调缔约双方协商的结果, 表示双方当事人对合同条款已经达成合意. 由此可见, “订”是“立”的过程, “立”是“订”的结果.

合同订立采用要约–承诺制度. 要约是一方当事人以缔结合同为目的, 向对方当事人提出合同条件, 希望对方当事人接受的意思表示. 发出要约的一方称为要约人, 接受要约的一方称为受要约人. 承诺是受要约人按照所指定的方式, 对要约的内容表示同意的一种意思表示. 采用“要约–承诺”制度优点是使合同成立过程清晰, 易于判断; 也有助于分清合同订立过程中双方的权利义务与责任.

解释 2: 智能合约归属

我国《电子签名法》第 2 条规定: “本法所称数据电文, 是指以电子、光学、磁或者类似手段生成、发送、接收或者存储的信息”.

智能合约采用计算机代码的形式表达合约条款, 它通过电子化方式被发送至区块链网络, 并被网络中所有节点接收和存储[23], 符合我国《电子签名法》的规定, 应被认定为数据电文.

其次, 智能合约以区块链为依托平台[24], 当事人可通过电子数据交换形式从区块链上随时调取查看合约内容, 并能以屏幕显示或打印形式, 有形地表现所载内容, 根据我国《民法典》第 469 条规定: “当事人订立合同, 可以采用书面形式、口头形式或者其他形式. 书面形式是合同书、信件、电报、电传、传真等可以有形地表现所载内容的形式. 以电子数据交换、电子邮件等方式能够有形地表现所载内容, 并可以随时调取查用的数据电文, 视为书面形式”, 因此属于数据电文的智能合约是书面形式, 其归属图如图 1 所示. 因此, 属于书面形式的智能合约在订立方面应符合《民法典》中的相关规定. 其中, 《民法典》第 471 条规定: “当事人订立合同, 可以采取要约、承诺方式或其他方式”, 为符合上述规定, 本文的智能合约订立过程采用亦要约–承诺方式. 注意: 未加说明的情况下, 合同是指法律上的传统合同, 合约是指智能(法律)合约的缩写, 泛指具有合同性质的代码化程序.

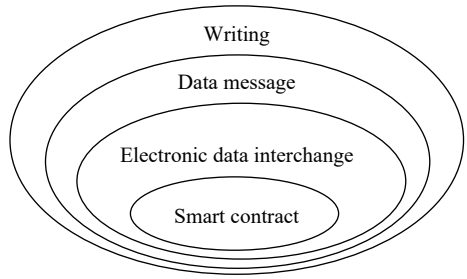

图 1　智能合约归属图

Fig.1　Venn diagram of a smart contract

3　系统框架

3.1　系统目标

缔约双方约定采用智能合约的形式订立合同，则合约订立应符合《民法典》等现行法律法规要求的成立规则．针对这一现实需求，本文将对基于区块链的智能合约系统进行合约订立设计，使得合约订立过程遵循现行法律规定，使区块链智能合约能够成为一种具有法律效力或法律意义的文书．

依据上述目标，本文对合约订立设计过程提出以下要求：

（1）订立流程合法化：从法律上规范化智能合约订立流程；

（2）意思表示真实性：在要约–承诺阶段，要约人和受要约人通过明示方式作出其意思表示；

（3）合同生效规范化：明确要约、承诺生效时间；

（4）合同存证合法化：对订立过程中的合约原件及数据进行合法存证．

3.2　合约模板化

智能法律合约是对同一类纸质合同经模板化后的电子化表示，也被称为合约示范文本（简称范本 Pattern 或模板 Template）．合约范本是一类合约实例的抽象化[25]，它包含格式条款和法律构成要素的属性，其中每个要素属性都有其唯一标识和类型约束．这里，属性值在合约模板中事先不必赋值，但经双方当事人协商、合约订立后需被确定．合约范本也符合《民法典》第 470 条："当事人可以参照各类合同的示范文本订立合同"的规定．

智能法律合约中条款属于格式条款．《民法典》第 496 条指出"格式条款是当事人为了重复使用而预先拟定，并在订立合同时未与对方协商的条款"．为了便于采用计算机处理合约中的格式条款，通常采用高级智能合约语言对其描述形成合约范本．此外，根据不同应用场景的实际需求，智能法律合约作为合约范本要为当事人提供书面化交互接口（见第 5.3 节），通过交互过程确定上述要素属性的取值，这一过程也被称为合约范本的实例化过程，所得结果被称为合约实例（Instance）．

3.3　智能合约订立框架

基于区块链的智能合约系统，本文进行"要约–承诺"制度的合约订立流程设计如图 2 所示．该订立框架包含如下实体：

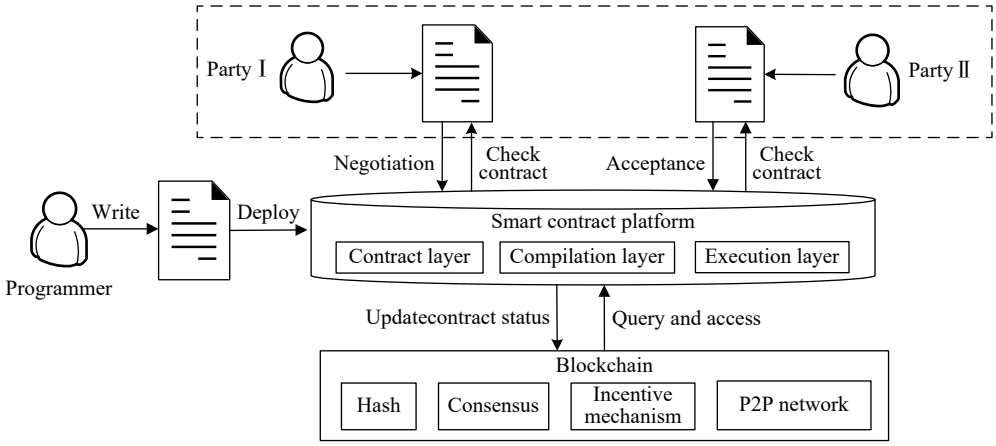

图 2　智能合约订立框架

Fig.2　Framework of smart contract conclusion

（1）甲方、乙方及编程人员：假定合同由甲乙双方当事人订立，他们均为具有相应民事权利能力和民事行为能力的人[26]．编程人员遵照商业规则采用智能合约语言撰写合约，并将其部署至智能合约平台．甲方通过平台调取查看合约后，如同意合约中条款表述则主动触发签名机制，以明示方式作出其意思表示，明确表明甲方已阅读、理解并同意本合约中的所有条款．乙方同样获取从智能合约平台返回的作为甲方"要约"的合约，若也同意合约中的权利义务关系，则也主动触发签名机制，以明示方式作出承诺，合约成立．

（2）智能合约平台：是一种支持智能合约可执行程序部署、签名、运行、验证的信息网络系统．包含合约层、编译层和执行层[27]，其中：

（a）合约层为编程人员提供智能合约编程语言、合约模板及与区块链交互的 API 接口；

（b）编译层将智能合约代码编译为虚拟机执行的字节码；

(c)执行层利用链上数据判断是否满足合约条款, 若满足则自动执行合约.

(3)区块链: 为智能合约提供了一个强有力的底层介质[28], 用于记录合约的代码、执行的中间状态及执行结果. 当前智能合约平台已经能够屏蔽区块链中的很多技术细节, 使得区块链中的各种复杂(哈希、P2P、共识机制、激励机制)机制为智能合约生命周期中的数据存证提供保障[29].

上述框架中甲、乙双方事先并未达成合意, 编程人员直接通过智能法律合约语言编写合约范本后, 智能合约程序被自动部署至智能合约平台, 双方当事人从平台调取查看合约内容, 如同意此合约中表述的权利义务关系, 则选择进行交易. 通过该方式也可反映出不同缔约主体间的合意, 自合约成立后, 双方当事人均受该意思表示约束.

4　智能合约订立流程合规化方案

类似于传统纸质合同的签订方式, 智能合约采取电子化形式进行要约–承诺认定, 双方当事人签署数字签名后即视为缔约双方对智能合约代码所表示条款的认可, 合约生效. 智能合约订立流程如图 3 所示, 包括智能合约从建立、部署、签名和存证四个阶段的处理.

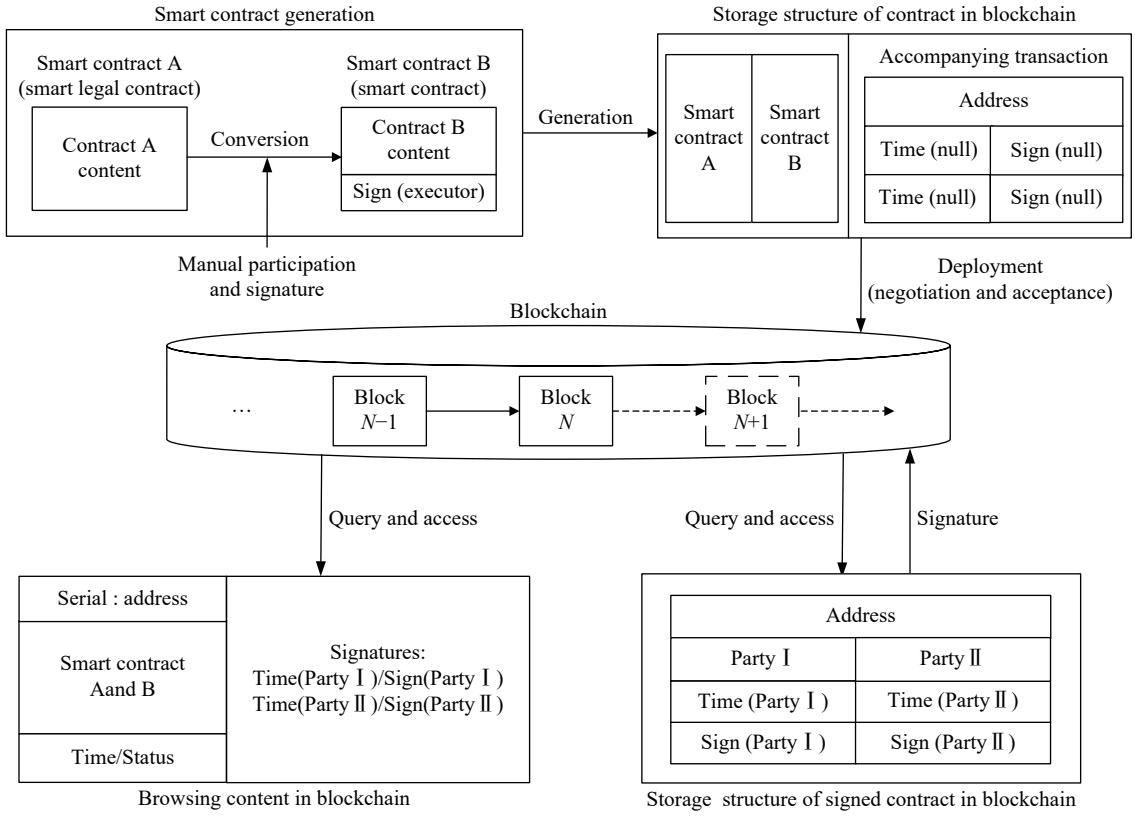

图 3　智能合约订立数据流程图

Fig.3　Data flow diagram of smart contract conclusion

4.1　智能合约建立

智能合约建立阶段是指编程人员撰写智能法律合约, 经一定转化规则生成计算机可执行程序[30]. 具体如下: 编程人员将缔约双方所描述的权利义务关系采用智能法律合约语言(如 SPESC 语言[12])撰写成智能法律合约, 即智能合约 A. 智能法律合约是对传统纸质合同的代码化后的结果, 将自然语言描述的合同条款用智能法律合约语言表述后, 可使合同条款在意思表示上更加精准简洁, 无二义性[31]. 经翻译后的智能法律合约与纸质合同相比, 虽然合同内容的载体不同, 但这并不影响缔约双方的合意呈现. 其次, 将智能合约 A 通过包含一定转化规则的合约翻译器转化为智能合约程序, 即智能合约 B. 例如, 文献 [17] 提出了一种从智能法律合约语言 SPESC 转化到以太坊智能合约语言 Solidity 的转化规则, 该转化规则的制定使得转化后的智能合约 B 具有规范的逻辑表达和函数结构, 避免了同一份智能合约 A 经不同编程人员转化后的不确定性.

智能法律合约(智能合约 A)到智能合约(智能

合约 B)转化过程通常应满足以下要求：

(1)证智能法律合约与转化后的智能合约具有相同的意思表示，具备相同的法律效力[32]；

(2)采用自动转化方式，转化在逻辑上是一种映射，保证了转化结果无二义性，原因在于所涉及的转化规则是确定的，从而使智能法律合约被转化后的结果是相同的；

(3)如果无法完成全部智能法律合约的自动转化，则允许人工参与．不同的编程人员对同一条款的解读和代码实现可能是不同的，但代码的执行结果必须是一致的．

在人工转化过程中，编程人员或法人必须对转化后的智能合约进行签名，并承担所编写代码引发问题的法律责任．如当事人对转化后的合同条款有争议，应根据《民法典》第 466 条第 1 款规定："合同文本采用两种以上文字订立并约定具有同等效力的，对各文本使用的词句推定具有相同含义．各文本使用的词句不一致的，应当根据合同的相关条款、性质、目的以及诚信原则等予以解释"．确定争议条款的意思表示．总之，上述过程无论是自动转化或是需人工参与，都必须保证智能合约 A 和转化后的智能合约 B 具备相同的意思表示．

4.2　智能合约部署

智能合约部署是指编程人员将智能合约 A 与智能合约 B 整合后部署至区块链智能合约平台的过程[30]．按照"要约–承诺"制度，合约部署后同意该合约的当事人才能进入合约订立阶段，因此，在智能合约部署过程中不仅涉及智能合约存证与可执行代码上链，还要为其后的智能合约订立预留接口．

为使区块链交易结构符合智能合约的订立要求，需要将已部署智能合约中的部分信息(如 4.2 节智能法律合约中的法律要素属性)分离出来并以交易形式独立进行存储，这种新的交易形式被称为伴随交易，它通常包含以下两类信息：

(1)智能合约范本中尚未确定并待当事人商议后确认的合约意思表示，比如，承诺生效时间、标的价格、标的物编号、付款方式等；

(2)当事人订立合约中"立"阶段的智能合约署名信息，比如，要约人和受要约人的信息、签署时间、数字签名等．

表 1 给出了一个伴随交易的示例结构．它是在比特币的交易结构上添加新交易字段加以构造，具体如下：

表 1　伴随交易的交易结构

Table 1　Structure of accompanying transaction

Notation		Description
vin[] ContractInput	contractAddress	Deployment address of contract
	latestCodeID	Transaction ID of latest contract code
	latestExecuteID	Transaction ID of latest contract status
	address	Contract address
	method	Contract interface
vout[] ContractOutout	contractData	Contract status
	listSign	Signature list
	signature	Signature of current executor
	signdate	Signature date of the current signer

(1)在输入(vin)字段中添加了 ContractInput 字段，该字段包含：系统自动部署智能合约可执行代码后获得的合约地址 contractAddress、最新合约代码的交易标识 latestCodeID、最新合约执行状态的交易标识 lastestExecuteID、当前执行合约的账户地址 address、当前所触发的合约接口 method；

(2)在输出(vout)字段中添加了 ContractOutput 字段，该字段包含：智能法律合约中的法律要素属性信息 contractData、合约签名列表 listSign、当前缔约方签名 signature、当前缔约方签名时间 signdate．

4.3　智能合约订立

智能合约订立是指能够使合约合法成立的过程．基于区块链的智能合约平台，要约人和受要约人需遵循相关法律要求的"要约–承诺"制度，使合约订立流程合法合规化．

基于区块链的智能合约平台为缔约双方提供合约范本库，双方可根据自身需求对合约范本进行选择．为使合约订立过程中的要约–承诺阶段能够满足现行法律对要约–承诺的认定，合约范本必须为当事人提供书面化交互接口．在"订"阶段，合约中任何需由当事人确定的信息，都必须由当事人经协商后主动填入．在"立"阶段，对合约的签名动作，必须保证由要约人/受要约人主动激活．因此，在合约订立交互接口中必须包含当事人主动注册为要约人/受要约人的接口(registPublish)及当事人主动对合约发起签名的接口(toSign)．

区块链以伴随交易的形式对合约订立过程中的交互数据进行存证．要约人(PartyI)阅读，理解并同意合约中所有表述，则通过主动触发方式激活合约范本中的签名交互接口，进而调用已部署

的智能合约可执行算法予以处理, 处理完毕后将签名时间 Time(PartyI) 和签名 Sign(PartyI) 以伴随交易的形式存储于区块链, 并将处理结果填充回合约范本, 供相关人员查看.

要约生效后, 要约过程中的所有数据都被存储于区块链, 任何时候相关人员都可请求查看合约内容. 受要约人(PartyII)若同意此要约, 则主动触发合约交互接口进行签名, 待已部署的智能合约可执行算法处理完毕后, 将签名时间 Time(PartyII) 和签名 Sign(PartyII) 同样以伴随交易的形式存储于区块链, 并将处理结果填充回合约范本以供查看.

4.4 智能合约存证

缔约双方经智能合约订立阶段后, 合约生效. 预先被存在链上的智能合约代码被当事人触发, 通过网络中多节点共识后, 按照合约中表述的条款自动执行. 智能合约从一个状态转变成另一个状态, 基于区块链的智能合约平台将合约代码、合约执行的中间状态 Status、执行结果以伴随交易的形式存储至区块链. 该过程可对合约订立及合约执行中合约状态的改变进行存证[30], 其中区块链一方面保证这些数据不被篡改, 另一方面通过每个节点以相同的输入执行智能合约来验证运行结果的正确性.

缔约双方及相关人员可通过交易 id 随时请求调取查看合约的代码、执行状态以及执行结果. 智能合约平台将智能法律合约和链上存储的交互数据进行整合, 并以书面化形式予以呈现.

总之, 上述智能合约订立方案遵循“要约–承诺”制度, 能够符合现行法律法规规范, 从而保证智能合约依法成立与法律效力.

5 智能法律合约的订立方案

第 4 节已从建立、部署、订立和存证四个阶段规范了智能合约订立流程. 此后我们将对该流程中的订立阶段进行详细设计, 使该阶段满足相关法律规定的“要约–承诺”制度, 保证合约合法成立.

智能法律合约作为法律合同转化为智能合约程序的过渡形式[33], 也需要遵循我国相关法律法规对合约订立的要求. 同时, 考虑到智能法律合约语言是编写智能法律合约的原则和依据, 因而首先需要在智能法律合约语言设计中满足合约订立的法律要求. 为了达到这一目的, 本节将首先以 SPESC 语言为例介绍智能法律合约, 继而在其中加入合约订立相关语法并加以示例.

5.1 智能法律合约语言

智能法律合约语言是一种面向法律合同领域的编程语言, 属于领域特定语言 DSL 的一种, 其特征就是符合现行法律规范. 文献 [12] 提出的智能合约规范化语言 SPESC 就是这类语言的一种. 采用该语言所撰写的智能法律合约有利于不同领域(法律、计算机及相关应用领域)人员协同设计和开发智能合约, 是一种更加高级的智能合约语言形式.

表 2 展示了 SPESC 的智能法律合约语法规则. 依据《民法典》第 470 条规定, 智能法律合约涵盖的合同内容包括: 当事人信息、标的、数量、质量、价款或者报酬、履行期限和方式、违约责任、解决争议的方法等方面, 因此, SPESC 语言编写的智能法律合约由合约框架(Contract)、合约名称(Title)、当事人描述(Parties)、标的(Assets)、资产表达式(AssetExpressions)、合约条款(Terms)、附加信息(Additions)等法律构成要素组成.

标的是指当事人权利和义务共同指向的对象, 以资产加以表示. 资产表达式则是智能合约语言中条款调用资产的形式, 通常涉及的资产操作包括存入(Deposits)、取回(Withdraws)、转移(Transfers)三类. 文献 [34] 对各种类型标的物(实物资产、虚拟资产、货币资产等)及其操作进行了全面阐述.

合同的主体是合同的各项条款, 也是确定当事人权利和义务的根据. 在 SPESC 语言中, 条款包括一般条款(GeneralTerms)、违约条款(BreachTerms)、仲裁条款(ArbitrationTerms)三种类型. 文献 [17] 中将合约名称、当事人描述、合同条款及附加信息进行了详细介绍, 并给出了由 SPESC 撰写合约转化为智能合约程序的方法. 然而, 上述工作都不涉及合约订立过程, 也没有在语言中支持合约订立相关语法.

5.2 智能法律合约中订立语法

在现行法律法规中合同订立法律构成要件基础上, 本文对智能法律合约语言进行了扩充并引入了合约订立(Contract conclusions)相关语法. 首先, 鉴于要约、承诺是合同成立的基本规则, 因此合约订立语法需体现要约和承诺两个阶段; 其次, 订立过程必须确认当事人的姓名或者名称和住所; 此外, 《民法典》第 490 条规定了采用合同书形式订立合同的, 自当事人均签名时合同成立, 因此, 智能法律合约必须支持当事人以数字签名形式对合约进行签名.

鉴于以上三点要求, 参照现有书面合同样式,

表 2　以 SPESC 为例的智能法律合约语法模型

Table 2　Grammar model of smart legal contract based on SPESC

Contract module name		Module description	Grammar definition
Contract framework		*Contract* ::= *Title*{ *Parties*+ *Assets*+ *Terms*+ *Additions*+ *Signs*+}	
Contract title		The contract title may consist of the of the contract name and the contract number.	*Title* ::= contract Cname (:serial number Chash)?
Contract party		The description of contract party can include the party's name, address, account number, and other attributes and values owned by this party, and technical measures such as the party's identity authentication can be adopted to ensure the uniqueness of its identity.	*Parties* ::= party group? Pname {*field*+ }
Contract subject		The contract subject refers to the object that the rights and obligations of the parties point to together, which is generally divided into things, behaviors, intellectual achievements, etc. Contract subject is represented by asset in the smart legal contract, and the description of such assets should exist in the blockchain.	*Assets* ::= asset Aname{ info{ *field*+ } right{ *field*+ }}
Asset expression		The asset expression used for the reference of a certain asset in contracting terms.	*AssetExpressions* ::= $ (amount)? (right of)? Aname
Asset operations	Deposit	The party can deposit assets voluntarily from his party account to contract account. The deposit operation is applied into the transaction of action execution in the term, in which the party can designate the deposited assets directly by asset expression, and restrict the assets through comparing two values by relational operation to determine their relationship. The latter is used to grant the permission for transferring the designated assets only if the relationship is satisfied.	*Deposits* ::= deposit (value RelationOperator)? *AssetExpression*
	Withdraw	The party can withdraw assets from contract account in the execution of terms, where the assets are designated by the asset expression.	*Withdraws* ::= withdraw *AssetExpression*
	Transfer	The party can transfer assets from contract account to other party account in the execution of terms.	*Transfers* ::= transfer *AssetExpression* to target
Contract terms	General terms	General terms include the term's name, the term's parties, the rights and obligations of the parties (actions that must, can or are prohibited), the execution conditions of the term, the asset transactions, and the post-conditions to be satisfied after the execution of the term.	*GeneralTerms* ::= term Tname: Pname (must|can|cannot) action(*field*+) (when preCondition)? (while transactions+)? (where postCondition)?.
	Breach terms	Breach terms refer to the legal liability to be assumed when the parties agreed by both parties do not perform the obligations stipulated in the smart legal contract or perform the obligations that do not conform to the contract. When the post-condition of the designated terms is not satisfied and the pre-condition of the breach term is satisfied, the related party must or can take action for setting the defaults, which may require to enforce asset operations and satisfy the post-condition of the breach term for the result of enforcement.	*BreachTerms* ::= breach term Bname (against Tname+)? : Pname (must|can) action(*field*+) (when preCondition)? (while transactions+)? (where postCondition)?.
	Arbitration terms	Arbitration terms stipulates the method to solve controversy in smart legal contracts. The specific controversy can be stated by nature language and an arbitration institution may be designated.	*ArbitrationTerms* ::= arbitration term : (The statement of any controversy)? administered by institution : instName.
Additional information		Additional information can define necessary supplementary information in smart legal contract, including entity's attribute, contract object, the property and signature of guarantor, additional term, program variable, and the declaration of data structure.	*Additions* ::= *field* + | (addition Dname {*field*+})

合约订立需包括两个部分：

（1）当事人宣称（statement）：用于双方当事人以意思表示的形式对要约与承诺过程中事项进行表述；

（2）当事人签名（signature）：用于记录合约成立的时间、地点、当事人及签名等信息，包括：打印名（printed-Name）、法定代表人签字（signature）、以及签订日期（date）等.

根据上述两部分的描述，智能法律合约中合约订立语法如下：

Signs ::= Contract conclusions :

(The statement of all parties.)?

{ Signature of party Pname:

{ printed-Name（打印名）: string,

signature（法定代表人签字）: string,

Date（签订时间）: date, +

},+

}

其中，?表示前面部分为可选内容，Pname 为当事人描述中定义的当事人名称（见表 2 中定义）. 在当事人宣称部分可用自然语言对要约和承诺过程进行描述，例如：

（1）除非以书面形式并经双方签署，否则本合约不得以任何方式修改.

（2）通过数字签名，表明缔约双方都已经阅读、理解并同意本合同中的所有条款和法规等.

（3）双方当事人同意智能法律合约及其转化的智能合约，与现实法律合同具有同等的法律地位. 在 3）中对智能法律合约及所转化智能合约的法律

效力进行肯定.

5.3　智能法律合约示例

在上述智能法律合约语言中添加订立语法后, 我们给出了一个由智能法律合约语言所撰写的打印机销售合约范本, 如图 4 所示. 该法律合约范本包含以下四方面内容:

```
contract purchase{
    party Seller{
        account : [action: registPublish]
        deliver()
        collectPayment ()
    }
    party Buyer{
        account : [action: registPublish]
        order()
        confirmReceive()
     }
    asset Printer{ info{
        name:  [type: string]
        value: [type: money]
    } }
    term no1: Buyer can order
        while deposit $ Printer::value.
    term no2: Seller must deliver
        when within 7 days after Buyer did order.
    term no3: Buyer must confirmReceive
        when within 7 days after Seller did deliver.
    term no4: Seller can colletPayment
        when after Buyer did confirmReceive
        while withdraw $ Printer::value.
     Contract conclusion:
       − This contract may not be modified in any manner unless
         in writing and signed by both parties.
       − By signing this agreement, all parties agree to the terms
         as described above.
       − Both parties agree with conversion from this contract to
         computer programs on smart contract platform, and approve
         that the programs' implementation has the same legal effect.
    { Signature of party Seller :
       {  printed-Name: [type: string],
          signature:    [action: toSign],
          date:         [type: string]
       }
     Signature of party Buyer :
       {  printed-Name: [type: string],
          signature:    [action: toSign],
          date:         [type: Date]
       }
    }
}
```

图 4　PESC 编写的打印机销售合约示范文本

Fig.4　Pattern of a printer sales contract written by SPESC

(1)当事人信息(party): 打印机销售合约范本包含卖方(Seller)和买方(Buyer). 其中, Seller 中 account 属性用于记录卖方的账户地址, 同时声明了卖方可以执行的两个动作: 交付打印机(deliver)和选择支付方式(collectPayment); Buyer 中 account 属性用于记录买方的账户地址以及声明了买方可以执行的两个动作: 预定打印机(order)和确认收货(confirmReceive).

(2)标的信息(asset): 作为标的的打印机定义包含 string 类型的 name 属性和货币类型的 value 属性.

(3)条款(term): 打印机销售合约范本中包含 4 条条款, 条款 No1 和条款 No3 定义了买方(Buyer)有权触发动作预定打印机(order)和确认收货(confirmReceive), 条款 No2 和条款 No4 定义了卖方(Seller)有权触发动作交付打印机(deliver)和选择支付方式(collectPayment).

(4)合约订立(Contract conclusion): 合约范本的当事人宣称部分包含了此类纸质合同所必需声明的意思表示; 双方签名部分包括两个 string 类型的 printed-Name 属性和 signature 属性, 及一个 Date 类型的 date 属性. 在图 4 合约实例中, 方框表

示部分为当事人提供书面化交互接口．以上述打印机销售合约范本为例，智能法律合约中方框所表示的两种类型的信息如下：

（a）要素属性信息：经双方当事人协商或由一方当事人填入后确定的信息．语法为“type: 属性类型”，type 是此处需填入的类型标志符．如打印机销售合约范本中，打印机定义的 name 属性为“type: string”，表示该属性须由卖方（Seller）填入的打印机型号，并且该属性值为 string；

（b）行为属性信息：由当事人主动触发动作后确定的信息．语法为“action: 动作名”，action 为动作标志符．如打印机销售合约范本中，卖方（Seller）所定义的 account 属性信息为“action: registPublish”，表示该属性是卖方（Seller）在主动触发动作 registPublish 时被填入的卖方账户地址（account）．签名时，卖方（Seller）和买方（Buyer）主动触发动作 toSign，将各自的打印名（printed-Name），签名（signature）和签名时间（date）填入．registPublish 动作和 toSign 动作的详细分析见 7.2 节．

上述打印机销售合约范本被定义完成后，双方当事人通过协商确定范本中的要素属性值（包括双方签名）完成合约订立，并发送给区块链对要素属性值进行存证．任何时候相关方都可查看订立后合同，并由智能法律合约平台在合约范本中添加区块链存证的要素属性值生成该合约实例（即订立后合同）．

6　转化后智能合约订立方案

6.1　合约订立流程

在 5.3 节智能法律合约语言编写的销售合约范本基础上，本节将给出该范本的合约订立方案．遵照合约订立“要约–承诺”的法律制度，销售合约通常将卖方 Seller 作为要约人，买方 Buyer 作为受要约人，因此，要约过程可视为卖方签署合约并向买方提供合同文本表明缔结合同的请求，承诺过程则视为买方接受请求并签署合约的行为．

销售合约订立流程如图 5 所示，详细步骤如下：

（1）将合约部署至基于区块链的智能合约平台，生成签名列表并以交易形式存储于区块链；

（2）执行方触发动作 registPublish，从区块链上下载签名列表，判断执行方身份？

（a）如果是要约人，则完成要约人注册并将其加入签名列表；要约人触发动作 toSign，执行要约人签名，并转至步骤（4）；

（b）如果不属于要约人，则转为步骤（3）；

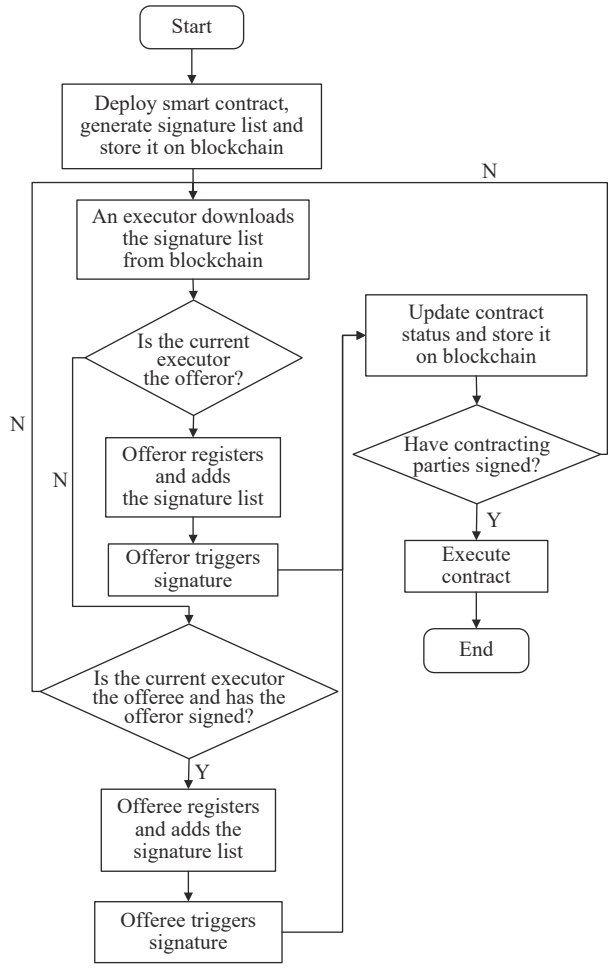

图 5　销售合约的订立流程图

Fig.5　Flowchart of sales contract conclusion

（3）判断当前执行方是否为受要约人且要约人已签名？

（a）如果是受要约人，则完成受要约人注册并将其加入签名列表；受要约人触发动作 toSign，执行受要约人签名，并转至步骤（4）；

（b）如果不是受要约人，则转为步骤（2）；

（4）填充签名信息至签名列表，并将更新后的签名列表发布至区块链；

（5）判断缔约双方是否均已签名？

（a）如果均已签名，进入合约执行阶段，程序结束；

（b）否则转至步骤 2；

在上述流程中必须保证要约人先对合约内容进行确认并签名，受要约人才被允许进行确认并签名．在具体实施中，要约人和受要约人都将通过动作触发的方式激活图 5 中方框内的行为属性，进而触发已部署的智能合约算法（registPublish 和 toSign）予以处理，并将结果填充回方框内供缔约双方查看．

6.2　智能合约中代码实现

缔约方触发动作 registPublish, 完成注册并加入签名列表, 注册流程如算法 1 所示. 该算法需输入当前缔约方账户地址 address、系统自动部署智能合约可执行代码后获得的合约地址 contractAddress 和一个布尔类型的 bOfferor(表示动作 registPulish 是否由要约方触发). 输出 listSign 数组, 用来存储 PartiesSigns 签名方类对象, 该对象表示合约订立语法的签名部分, 包括三个基本属性: 签名方地址、对应签名和签名时间. 其中, 算法中的判断条件 signUser==NULL 是防止缔约方重复注册签名列表. 在合约中, 可以对签名方进行添加操作, 签名方账户地址作为当事人的唯一标识.

算法 1 **registPublish**

Input: address, contractAddress, bOfferor
Output: flag
flag ← false;
listSign ← getListSign(contractAddress);
signUser← listSign.getSignUser(bOfferor);
if (signUser==NULL)
  PartiesSigns ps = new PartiesSigs();
  ps.setUserAddress(address);
  ps.setUserbOfferor(bOfferor);
  listSign.add(ps);
  flag=true;
end if
return listSign;

缔约方触发动作 toSign, 进行签名, 完成合约订立, 订立流程如算法 2 所示. 该算法需输入当前执行方账户地址 address, 合约地址 contractAddress 和一个布尔类型的 bOfferor(表示动作 toSign 是否由要约方触发). 输出是一个布尔类型的 flag, 表示当前动作是否执行成功. 其中, 该算法的第一层判断条件 signUser!=NULL 是验证触发动作的执行方是否存在于签名列表; 第二层判断条件 signUser.getSignature() 是防止二次签名, bOfferor==bCheckOfferor 是保证触发动作的执行方与签名列表中的签名方是同一个人; 第三层判断条件 bOfferor==true 是判断当前触发动作的执行方是否是要约人, 如果 bOfferor 取值为 true, 表明当前执行方是要约方, 则直接调用 signMessage 函数进行签名, 否则执行判断条件 listSign.getOfferorSignature() != null 判断要约人是否已签名, 如果已签名, 则受要约人直接调用 signMessage 函数进行签名. signMessage 函数采用的是椭圆曲线数学签名算法 ECDSA-secp256k1.

算法 2 **toSign**

Input: address, contractAddress, bOfferor
Output: flag
flag ← false;
listSign ← getListSign(contractAddress);
sreMessage ←getMessage(contractAddress);
signUser← listSign.getAddress(address);
if (signUser!=NULL)
  bCheckOfferor ←signUser.getbOfferor();
  if (signUser.getSignature()==NULL && bOfferor==bCheckOfferor)
    if(bOfferor==true||listSign.getOfferorSignature() != null)
      signature ← signMessage(address, strMessage);
      date ← currentTime();
      signUser.setSignature(signature);
      signUser.setSignDate(date);
      signUser.setSignAddress(address);
      flag ← true;
      end if
  end if
end if
return flag;

7　合约实例

在上述合约订立过程中区块链负责对交互数据进行存证, 这些存证应符合我国现行法律的存证要求, 同时, 区块链不可修改的特性也保证了存证的真实性[35]. 根据合约订立过程, 我们不难发现一个合约实例由其对应的范本及交互信息共同生成, 这就保证了同一个合约范本能够支持大量的合约实例. 为了符合上述合约订立的存证要求, 本节将以 7.1 节销售合约的订立过程为实例, 对所涉及的区块链存证方案进行设计, 并通过实验案例加以验证.

实验采用基于 Bitcoin 架构的区块链系统. 该系统以 JSON 格式的伴随交易(Transaction)(见表 1)为数据结构进行数据存储, 其中 JSON 是一种采用“键值对”形式存储数据, 并支持数组([…]表示)和集合({…}表示)两种结构. 下面将基于 JSON 格式对智能合约执行流程所涉及的交易结构进行介绍:

(1)注册合约的交易结构: 图 6 为买方 Buyer 触发动作 registPublish 的注册交易结构, 包含输入

字段和输出字段．输入字段有 method、params 等．其中，method 是本次触发合约的动作名，params 为参数数组，第一个参数是买方 Buyer 的账户地址，第二个参数是合约地址．输出字段有 listSign, txid．其中 listSign 为更新后的签名方列表；txid 表示此次注册合约的交易 id，通过该字段可恢复此次合约状态．该触发动作被已部署的智能合约算法 registPublish 处理完毕后，将买方 Buyer 账户地址填充回买方 Buyer 的 account 属性方框内，供缔约双方查看．

```
{
    method:"registPublish",
    params:["1EPnLUmG4o……bwPizGr4L",
            "1TttnpFTcc……TtVDesSqSRf"],
    id:1,
    chain_name:"testchain"
}{
    listSign:"fkoer456……536hyio817",
    txid:"2a0b2c15……6b631844478fcbd5"
}
```

图 6　买方 Buyer 注册合约交易结构图

Fig.6　Transaction datagram of a buyer registration contract

（2）订立合约的交易结构：缔约方通过触发动作 registPublish 后将其加入到 listSign 签名列表中，等待合约签名．图 7 为买方 Buyer 有权触发签名后的订立交易结构．输入字段与上述注册交易的输入字段一致．输出字段包含的 4 个属性：userResult 用于提示签名方是否签名成功；listSign 为签名完成后更新的签名列表；signature 为当前执行方签名；signdate 为签名时间；txid 为此次合约订立的交易 id，通过该 id 可恢复此次合约状态．该签名触发动作被已部署的智能合约算法 toSign 处理完毕后，将签名 siganture 和签名时间 signdate 填充回买方 Buyer 的合约订立签名部分．

```
{
    method:"toSign",
    params:["1EPnLUmG4o……bwPizGr4L",
            "1TttnpFTccHjq……LUsSqSRf"],
    id:1,
    chain_name:"testchain"
}{
    userResult:"Sign Success!!!,but not all",
    listSign:"aced0013……1736a656a78",
    signature:"HR5pkp0vEE……MpP7g0=",
    signdate:"2020-10-26 10:29:58",
    txid:"f48ecfd45bbc3db5……df60820cd"
}
```

图 7　买方 Buyer 签名交易结构图

Fig.7　Transaction datagram of a buyer signature contract

（3）执行合约的交易结构：以销售合约中条款 No1 执行为例，图 8 为买方 Buyer 触发动作预定打印机 order 的执行合约交易结构，包含输入字段和输出字段．其中，输入字段中 method 为本次触发的合约动作名；params 为合约参数数组，第一个参数为买方 Buyer 的账户地址，第二个参数为合约地址，第三个参数为买方 Buyer 所传入的预定金额．输出字段中 userResult 为预定提示信息，txid 为此次执行合约的交易 id．

```
{
    method:"order",
    params:["1EPnLUmG4o……bwPizGr4L",
            "1TttnpFTccHjq……LUsSqSRf",
            "2500"],
    id:1,
    chain_name:"testchain"
}{
    userResult:"Order Successfully",
    txid:"38ca12ac8fed3fe……5a8d09c44278"
}
```

图 8　买方 Buyer 订购交易结构图

Fig.8　Transaction datagram of a buyer purchase contract

上述交易结构的设计使得合约注册，订立和执行三个阶段的交互数据都被存证，缔约双方及相关人员可通过交易 id 随时请求调取查看合约内容．基于区块链的智能合约平台将智能法律合约和链上存储的交互数据进行整合，并以书面化形式予以呈现．图 9 是 5.3 节销售合约范本与链上交互数据整合后的合约实例书面化呈现．

8　方案合规性辩析

本节将对第 6 节智能法律合约订立方案的合规性进行辨析，从而论证所提方案能够满足我国现行法律的规定，由该方案处理的智能合约具有法律效力．

辨析 1：本文合约订立过程中的要约阶段满足现行法律对要约的认定．

《民法典》第 472 条规定：“要约是希望与他人订立合同的意思表示，该意思表示应当符合下列条件：内容具体确定；表明经受要约人承诺，要约人即受该意思表示约束”．据此，智能法律合约订立中的要约阶段需满足以下要求：

（1）合约内容具体确定．

智能法律合约作为法律合同转化为智能合约程序的过渡形式，同样以数字代码的形式呈现．如 5.3 节所示，从 SPESC 撰写的打印机销售合约范本中可得出如下事实：首先，SPESC 语言作为计

```
contract purchase{
    party Seller{
        account: 1aAeGrzMfm……HaAdtyhp3
        deliver()
        collectPayment ()
    }
    party Buyer{
        account: 1EPnLUmG4o……bwPizGr4L
        order()
        confirmReceive()
    }
    asset Printer{ info{
        name:  "Printer"
        value:  2500 RMB
    }  }
    term no1: Buyer can order
        while deposit $ Printer::value.
    term no2: Seller must deliver
        when within 7 days after Buyer did order.
    term no3: Buyer must confirmReceive
        when within 7 days after Seller did deliver.
    term no4: Seller can colletPayment
        when after Buyer did confirmReceive
        while withdraw $ Printer::value.
    Contract conclusion:
        - This contract may not be modified in any manner unless
          in writing and signed by both parties.
        - By signing this agreement, all parties agree to the terms
          as described above.
        - Both parties agree with conversion from this contract to
          computer programs on smart contract platform, and approve
          that the programs' implementation has the same legal effect.
  {  Signature of party Seller :
    {   printed-Name:   Yao San
        signature:      Ye23de198R……Gelkap3
        date:           2020-7-12 15:37:45
    }
    Signature of party Buyer :
    {   printed-Name:   Wan Liu
        signature:      HR5pkp0vEE……MpP7g0=
        date:           2020-10-26 10:29:58
    }
  }
}
```

图 9　被填充后的打印机销售合约示范文本

Fig.9　Filled pattern of a printer sale contract

算机语言具有明确形式化、无二义性语法, 满足《民法典》规定合同内容显式定义的要求; 其次, 该合约范本中条款包含了权利义务关系的声明, 进一步使要约人的意思表示内容具体确定; 此外, 该合约范本中订立语法不仅包含了由缔约双方不断协商后所约定的意思表示, 也包含了当事人姓名、签名、合约成立时间等信息. 因此, 由智能法律合约语言所撰写的智能法律合约其内容具体且确定.

(2)要约人受该意思表示约束.

智能法律合约作为合约范本, 为当事人提供了书面化交互接口, 如 5.3 节的销售合约范本. 合约内容以书面化形式供缔约双方阅读, 双方通过不断协商确定法律合约中要素属性信息的值, 并通过动作激活行为属性, 进而触发智能合约可执行代码予以处理. 上述一切的交互形式都是缔约双方在阅读, 理解并同意合约中所有条款的情况下主动发起的.

合约范本订立语法中包含当事人宣称部分. 该部分是缔约双方对要约和承诺过程中的事项进行不断协商后的表述, 因此, 这部分是用户主观的意思表示, 如 5.3 节销售合约范本的订立宣称部分包含如下约定: “双方当事人同意高级智能法律合约及其转化的可执行程序, 与现实法律合同具有同等的法律地位”. 即表示缔约双方肯定了智能法律合约的法律地位, 愿意同纸质合同一样受其约束.

《民法典》第 140 条第 1 款规定: “行为人可以明示或者默示作出意思表示”, 在本文订立语法的

当事人签名部分, 要约人通过明示方式作出其意思表示, 如主动触发上述行为属性信息, 明确表明要约人已经同意本合约中的所有内容. 在 5.3 节销售合约范本订立部分, 卖方 Seller 如同意该合约则主动触发动作 toSign, 对合约签名, 生成签名 signature 及签名时间 signdate.

因此, 智能法律合约经受要约人承诺后, 要约人必受该合约内容的意思表示约束.

(3)要约生效时间.

《民法典》第 137 条第 2 款规定: "当事人对采用数据电文形式的意思表示的生效时间另有约定的, 按照其约定". 区块链负责对合约订立过程中的交互数据进行存证, 要约人可查看合约内容, 如希望与他人缔结合同, 并已同意合约内容, 则主动触发动作 toSign 予以签名, 即要约, 因此可将要约生效时间约定为要约人签名时间. 如第 5.3 节销售合约范本中订立合约部分, 卖方 Seller 主动触发动作 toSign, 通过被已部署的智能合约算法 toSign 处理完毕后, 将签名 siganture 和签名时间 signdate 填充回卖方 Seller 的合约订立签名部分, 此时要约生效, 该签名时间为要约生效时间.

辨析 2: 本文合约订立过程中的承诺阶段满足现行法律对承诺的认定.

《民法典》第 479 条规定: "承诺是受要约人同意要约的意思表示". 据此, 智能法律合约订立中的承诺阶段需满足以下要求:

(1)受要约人意思表示.

如同上述要约认定第 2 点分析, 合约内容以书面化形式供缔约双方查看, 法律合约范本中所有的要素属性信息(包括签名部分)都是受要约人主动触发的. 同时, 受要约人也通过明示方式作出意思表示, 明确表明受要约人已经阅读、理解并同意本合约中的所有条款. 如第 5.3 节销售合约范本中订立合约部分, 买方 Buyer 主动触发动作 toSign, 被已部署的智能合约算法 toSign 处理完毕后, 将签名 siganture 和签名时间 signdate 填充回买方 Buyer 的合约订立签名部分, 即买方 Buyer 对该合约承诺, 从而证实受要约人接受该意思表示.

(2)承诺生效时间.

《民法典》第 484 条规定: "以通知方式作出的承诺, 生效的时间适用本法第 137 条的规定". 在本文的订立流程中, 要约生效后将要约过程中的交互数据发布至区块链, 受要约人可查看合约内容及交互数据, 若同意此要约, 则触发动作 toSign 予以签名, 此时表示受要约人已接受要约内容, 即承诺, 因此可将受要约人对其进行签名的时间视为承诺时间. 另据《民法典》第 483 条规定: "承诺生效时合同成立", 可知在销售合约范本中当买方 Buyer 对其进行承诺后, 合约成立.

辨析 3: 本文合约订立过程中对于合约原件及交互数据保存形式满足现行法律要求.

《电子签名法》第 5 条规定: "符合下列条件的数据电文, 视为满足法律、法规规定的原件形式要求: 能够有效地表现所载内容并可供随时调取查用; 能够可靠地保证自最终形成时起, 内容保持完整、未被更改". 合约订立是一个交互的过程, 区块链负责为交互过程中的数据进行存证. 因此, 本文合约订立过程需满足以下要求:

(1)随时调取查用.

一个合约实例由其对应的范本及交互信息共同生成, 为满足合约订立过程中对交互数据存证的需求, 在第 8 节我们设计了智能合约执行流程中所涉及的三种交易结构, 将当事人注册、签名及运行合约的全流程存储于区块链, 区块链的公开透明性使得任何时候智能法律合约范本都可从中获取到交互数据对要素属性值进行填充, 供相关方调取查用.

(2)存储可靠性.

智能法律合约存储的可靠性取决于区块链的存储可靠性. 区块链本身具有防篡改、难删除和公开透明的特性, 合约交易数据一经全网共识即被永久的存于链上[36]. 在数量庞大的节点中, 必须同时破坏 51% 的节点, 才会影响整个系统的运行, 篡改区块数据, 但这仅仅在理论上是可能的. 因此, 符合上述《电子签名法》第五条规定.

(3)形式完整性.

区块链中数据存储采用 JSON 格式, 尽管这种格式与书面化的智能法律合约不同, 但这并不影响智能法律合约的内容, 且其可随时准确恢复到原书面化形式. 根据《电子签名法》第 5 条规定: "在数据电文上增加背书以及数据交换、储存和显示过程中发生的形式变化不影响数据电文的完整性". 因此, 区块链存证并不影响智能法律合约的形式完整性.

综上, 本文在基于区块链的智能合约系统上提出的合约订立方案能够满足现行法律法规对要约、承诺的认定及对合约原件形式保存的要求.

9　结论

本文为使智能法律合约订立过程符合相关法

律所规定的要约–承诺制度, 对智能法律合约语言进行了扩充, 引入了合约订立相关语法和合约范本概念, 提出了基于数字签名的法律要约和承诺过程及相应算法. 同时, 为了支持上述方案的实现, 在区块链平台中设计了三种交易结构, 能够使合约订立过程中的交互数据存证满足现行法律存证要求. 本文所做工作将有助于为智能法律合约的订立过程提供法律依据, 促进我国智能合约的法律化建设.

参 考 文 献

[1] Szabo N. The Idea of Smart Contracts (1994) [J/OL]. *Public Networks*. http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

[2] Szabo N. Smart contracts: building blocks for digital markets [J/OL]. *Public Networks*.https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html

[3] Nakamoto S. Bitcoin: a peer-to-peer electronic cash system [J/OL]. *Bitcoin Online*.https://bitcoin.org/bitcoin.pdf

[4] Fan J L, Li X H, Nie T Z, et al. Survey on smart contract based on blockchain system. *Comput Sci*, 2019, 46(11): 1
(范吉立, 李晓华, 聂铁铮, 等. 区块链系统中智能合约技术综述. 计算机科学, 2019, 46(11): 1)

[5] Buterin V. A next-generation smart contract and decentralized application platform [J/OL]. *Ethereum White Paper*.https://translatewhitepaper.com/wp-content/uploads/2021/04/Ethereum-Orijinal-ETH-English.pdf

[6] Bertoli P. *Blockchain, Law and Governance*. Berlin: Springer International Publishing, 2020

[7] Governatori G, Idelberger F, Milosevic Z, et al. On legal contracts, imperative and declarative smart contracts, and blockchain systems. *Artif Intell Law*, 2018, 26(4): 377

[8] Kasprzyk K. The concept of smart contracts from the legal perspective. *Rev Eur Comp Law*, 2019, 34(3): 101

[9] Guo S F. Blockchain smart contracts in contract law. *Orient Law*, 2019(3): 4
(郭少飞. 区块链智能合约的合同法分析. 东方法学, 2019(3): 4)

[10] Chen J D. The legal structure of smart contract. *Orient Law*, 2019(3): 18
(陈吉栋. 智能合约的法律构造. 东方法学, 2019(3): 18)

[11] Farmer W M, Hu Q. A formal language for writing contracts // 2016 *IEEE 17th International Conference on Information Reuse and Integration* (*IRI*). Pittsburgh, 2016: 134

[12] He X, Qin B H, Zhu Y, et al. SPESC: A specification language for smart contracts // 2018 *IEEE 42nd Annual Computer Software and Applications Conference* (*COMPSAC*). Tokyo, 2018: 132

[13] Regnath E, Steinhorst S. SmaCoNat: smart contracts in natural language // 2018 *Forum on Specification & Design Languages* (*FDL*). Garching, 2018: 5

[14] Mavridou A, Laszka A. Designing secure ethereum smart contracts: A finite state machine-based approach // *Financial Cryptography and Data Security*. Berlin, 2018: 523

[15] Choudhury O, Rudolph N, Sylla I, et al. Auto-generation of smart contracts from domain-specific ontologies and semantic rules // 2018 *IEEE International Conference on Internet of Things* (*iThings*) *and IEEE Green Computing and Communications* (*GreenCom*) *and IEEE Cyber, Physical and Social Computing* (*CPSCom*) *and IEEE Smart Data* (*SmartData*). Halifax, 2018: 963

[16] Zupan N, Kasinathan P, Cuellar J, et al. *Blockchain Technology Industry* 4.0. Berlin: Springer International Publishing, 2020

[17] Zhu Y, Qin B, Chen E, et al. An advanced smart contract conversion and its design and implementation for auction contract. *Chin J Com*, 2021, 44(3): 1

[18] Clack C D, Bakshi V A, Braine L. Smart contract templates: foundations, design landscape and research directions [J/OL]. *arXiv*. https://arxiv.org/abs/1608.00771

[19] Clack C D, Bakshi V A, Braine L. Smart contract templates: essential requirements and design options[J/OL]. *arXiv*. https://arxiv.org/abs/1612.04496

[20] Clack C D. Smart contract templates: legal semantics and code validation. *J Digit Banking*, 2018, 2(4): 338

[21] Openlaw. OpenLaw's documentation [EB/OL]. https://docs.openlaw.io/

[22] Zhou X Z, Li Y. An analysis of legal issues in the formation of electronic contracts. *Forum Jiang Su Commer*, 2003(8): 120
(周显志, 李莹. 电子合同订立中的法律问题探析. 江苏商论, 2003(8): 120)

[23] Wang S, Yuan Y, Wang X, et al. An overview of smart contract: Architecture, applications, and future trends // 2018 *IEEE Intelligent Vehicles Symposium* (*IV*). Changshu, 2018: 108

[24] Bartoletti M, Pompianu L. An empirical analysis of smart contracts: Platforms, applications, and design patterns. *Financial Cryptogr Data Secur*, 2017: 494

[25] Wohrer M, Zdun U. From domain-specific language to code: Smart contracts and the application of design patterns. *IEEE Softw*, 2020, 37(5): 37

[26] Xu S. The contract attributes of smart contracts and their legal regulations. *J Heilongjiang Adm Cadre Coll Politics Law*, 2021(1): 74
(徐颂. 智能合约的合同属性及其法律规制. 黑龙江省政法管理干部学院学报, 2021(1): 74)

[27] Zhu Y, Wang Q S, Qin B H, et al. Survey of blockchain technology and its advances. *Chin J Eng*, 2019, 41(11): 1361
(朱岩, 王巧石, 秦博涵, 等. 区块链技术及其研究进展. 工程科学学报, 2019, 41(11): 1361)

[28] Watanabe H, Fujimura S, Nakadaira A, et al. Blockchain contract: Securing a blockchain applied to smart contracts // 2016 *IEEE International Conference on Consumer Electronics* (*ICCE*). Las

Vegas, 2016: 467

[29] Zheng Z B, Xie S A, Dai H N, et al. An overview of blockchain technology: Architecture, consensus, and future trends // 2017 *IEEE International Congress on Big Data* (*BigData Congress*). Honolulu, 2017: 557

[30] Chinese Institute of Electronics. T/CIE 159-2020 *Formal Expression of Blockchain Smart Contract*. Beijing: China Standard Press, 2020
（中国电子学会. T/CIE 095-2020区块链智能合约形式化表达. 北京: 中国标准出版社, 2020）

[31] Liu Q, Wang D J, Wang X X, et al. Review on conformance between legal contract and smart contract. *Appl Res Comput*, 2021, 38(1): 1
（刘琴, 王德军, 王潇潇, 等. 法律合约与智能合约一致性综述. 计算机应用研究, 2021, 38(1): 1）

[32] Savelyev A. Contract law 2.0: Smart contracts as the beginning of the end of classic contract law. *Inf Commun Technol L*, 2017, 26(2): 116

[33] Haapio H, Hagan M. Design patterns for contracts // *Networks. Proceedings of the* 19*th International Legal Informatics Symposium IRIS*. Wien, 2016: 381

[34] Zhu Y, Song W J, Wang D, et al. TA-SPESC: Toward asset-driven smart contract language supporting ownership transaction and rule-based generation on blockchain. *IEEE Trans Reliab*, 2021, 70(3): 1255

[35] Li J, Chen Y S, Song H T. Research on digital currency supervision model based on blockchain technology. *J Phys Conf Ser*, 2021, 1744(3): 032112

[36] Wang Y D, Li L, Hu D. A literature review of block chain. *J China Univ Min Technol* (*Soc Sci*), 2018, 20(3): 74
（王元地, 李粒, 胡谍. 区块链研究综述. 中国矿业大学学报(社会科学版), 2018, 20(3): 74）