工程科学学报
Chinese Journal of Engineering

智能法律合约及其研究进展

王迪 朱岩 陈娥 郭倩 李冀宁 孙贻滋 伊然

Smart legal contract and its research progress

WANG Di, ZHU Yan, CHEN E, GUO Qian, LI Ji-ning, SUN Yi-zi, YI Ran

引用本文:
王迪, 朱岩, 陈娥, 郭倩, 李冀宁, 孙贻滋, 伊然. 智能法律合约及其研究进展[J]. 工程科学学报, 2022, 44(1): 68-81. doi: 10.13374/j.issn2095-9389.2021.02.22.001
WANG Di, ZHU Yan, CHEN E, GUO Qian, LI Ji-ning, SUN Yi-zi, YI Ran. Smart legal contract and its research progress[J]. *Chinese Journal of Engineering*, 2022, 44(1): 68-81. doi: 10.13374/j.issn2095-9389.2021.02.22.001

在线阅读 View online: https://doi.org/10.13374/j.issn2095-9389.2021.02.22.001

工程科学学报, 第 44 卷, 第 1 期: 68−81, 2022 年 1 月
Chinese Journal of Engineering, Vol. 44, No. 1: 68−81, January 2022
https://doi.org/10.13374/j.issn2095-9389.2021.02.22.001; http://cje.ustb.edu.cn

智能法律合约及其研究进展

王　迪[1)], 朱　岩[1)✉], 陈　娥[1)], 郭　倩[1)], 李冀宁[2)], 孙贻滋[2)], 伊　然[3)]

1) 北京科技大学计算机与通信工程学院, 北京 100083　2) 中国电子学会, 北京 100036　3) 北京互联网法院, 北京 100160
✉通信作者, E-mail: zhuyan@ustb.edu.cn

摘　要　从智能合约、智能法律合约等概念入手, 依据现行法律条目的要求对智能合约法律化问题进行探讨, 指出智能合约法律化需满足文法要求、非赋权原则、审查准则三个基本规则, 并以典型智能法律合约语言 SPESC、CML 为实例剖析了其法律效力, 辨析使其与原合同文本具有同等法律效力需满足的条件. 进而, 结合智能合约系统架构及部署运行过程, 在对所部署智能合约进行法律化辨析基础上对区块链智能合约及其链码的法律地位进行了论证. 最后, 对当前智能法律合约逻辑模型与语言模型的研究进展进行总结, 并加以讨论和评价. 上述工作表明当前智能法律合约研究是一条解决智能合约法律地位的可行途径, 有利于从现行法上把握智能合约在合约逻辑、仲裁流程、形式化验证等方面的未来发展方向.

关键词　智能法律合约; 领域专用语言; 数据电文; 法律化原则; 区块链
分类号　TP319

Smart legal contract and its research progress

WANG Di[1)], *ZHU Yan*[1)✉], *CHEN E*[1)], *GUO Qian*[1)], *LI Ji-ning*[2)], *SUN Yi-zi*[2)], *YI Ran*[3)]

1) School of Computer & Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
2) The Chinese Institute of Electronics, Beijing 100036, China
3) Beijing Internet Court, Beijing 100160, China
✉ Corresponding author, E-mail: zhuyan@ustb.edu.cn

ABSTRACT　With the advancement of blockchain, smart contracts have become increasingly popular. However, the uncertain status by law severely limits their practical applications. To address the problem, smart legal contract (SLC) is proposed as a transitional technology between legal and smart contracts. Starting with the basic concept of SLCs, in this paper, we discussed the legalization of smart legal contracts based on the requirements of existing legislation items and highlight that legalization should meet three elementary principles, including the specified grammatical requirements (for regulating terminology and eliminating ambiguity), the principle of nonempowerment (for resolving the inherent contradiction between automatic execution and the rights of parties), and examination criteria (for handling legal validity and code security issues). Moreover, we analyzed SLC's legal effect by taking typical smart legal contract languages, SPESC and CML as examples, and show that the contract program or chaincode has the same legal effect as the original contract, if and only if they satisfy three necessary conditions: (1) adopting the technical specification for generation and conclusion of SLCs; (2) complying with three abovementioned elementary principles; and (3) agreeing on declaration with the same legal effect. Furthermore, investigating the smart contract architecture and deployment, the legal status of both contract program and compiled chaincode was demonstrated in legal analysis of the deployed smart contract. Last, we discussed and evaluated the current situation of smart legal contract logic models and language models on SLCs. This work shows that the research on smart legal contracts is a suitable approach to guarantee the legal status of smart contracts, and the results will contribute to grasping the future research directions in several fields, such as contract logic, arbitration process, and formal verification, from the existing legislation viewpoint.

收稿日期: 2021−01−22
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)

KEY WORDS　smart legal contract; domain specific language; data massage; principle of legalization; blockchain

区块链是采用密码手段保障、只可追加、链式结构组织的分布式账本系统[1]，其核心价值在于实现了多参与方在统一规则下的自发高效协作，并通过代码、协议、规则为分布式账本及其网络提供了信用基础. 智能合约（Smart contract）[2–3] 是第二代区块链的核心技术之一，允许开发者利用编程语言编写可自动执行程序，实现价值交换等应用并在区块链上存证.

智能合约是法律与计算机领域结合而提出的概念，为了进一步保证智能合约的法律地位，智能法律合约（Smart legal contract, SLC）[4–5] 被提出，其目的在于在现行法下将智能合约形式及其代码执行行为认定为法律意义上的电子合同. 智能法律合约与智能合约相比较，具有更加严格的法律特征，这表现在它的表达语言、表现形式、执行方法等各方面.

智能法律合约是智能合约法律化发展的必然结果. 现实生活中既懂法律又懂区块链领域专业知识的人仍然较少，使得审查计算机代码法律性的现实难度增加，智能法律合约的出现有助于降低这种难度. 同时，智能法律合约作为一种新的区块链应用软件开发模式，不仅扩大了区块链技术的应用领域和快速开发能力，也通过规则约束代码运行，为人机物之间的高效安全协作提供了技术保障，从而降低交易成本并减少法律纠纷.

从目前的工程实践和学术成果来看，当前智能法律合约概念与技术尚属于起步阶段，有待进一步提高，对其概念、法律地位及内涵均缺乏系统性的研究，同时，对面向智能法律合约的软件模型和编程语言、及运行环境也缺少体系上的归纳与总结.

有鉴于此，本文从智能法律合约的历史及概念演变过程入手，分析了当前智能合约平台及其构架，阐述了智能法律合约的法律化思考，并以近年来此领域的实践研究为基础，辨析了智能法律合约法律化地位需满足的基本规则，并指出了为使其与原合同文本具有同等法律效力需满足的3方面条件. 上述辨析与分析表明智能法律合约是一条保障智能合约法律地位的可行途径，有利于从现行法上把握智能合约在合约逻辑、仲裁流程、形式化验证等方面的未来研究方向.

1　智能合约概念

合约是当事人基于意思表示合致而签订的协议，是一个使未取得彼此信任的各参与方具有安排权利与义务的商定框架[6]；智能合约在广义上是指符合当事人之间约定的任何计算机协议，是一种计算机化的合约. 智能合约运行必须满足参与者事先的约定，且其计算机任务的完成需两或多名参与者共同协作，因此智能合约既能满足合约遵循的可信性与合规性（合法性），也为保证合约合规性提供了协议验证、存证、争议解决等必需的技术手段.

上述定义的内涵较为宽泛，可囊括大多数的计算机网络协议. 为了更加明确智能合约的法律化属性，维基百科中给出了如下定义[7]：

定义1 智能合约：智能合约是一种旨在根据合约或协议中条款自动执行、控制或记录与法律相关事件和动作的计算机程序或交易协议.

此定义体现了智能合约处理对象是法律合约，处理手段是计算机协议，该手段目的是促进、保障、验证、加强合约协商和履行[8]，表现形式是计算机程序或交易协议.

就区块链平台而言，区块链智能合约定义为：

定义2 区块链智能合约：区块链智能合约是部署在区块链上、在满足预定条件时可自动执行并存证的计算机程序[9].

与广义智能合约概念相比较，这种智能合约的载体是区块链，它本质是一种自动执行与存证的计算机程序. 通常，智能合约是平台无关的概念，而区块链智能合约是平台相关的，需针对特定区块链结构转化为该平台指令系统所理解的代码. 这种转化后的代码被称为链码（Chaincode），具体定义如下：

定义3 链码：链码是指由区块链智能合约转化的、部署在区块链中并可被直接执行的指令序列.

链码将合约条款内容转化的指令序列直接写入区块链代码行中，并设置代码执行的触发条件，因代码在满足预定触发条件后自动运行，不需要人为干预，因此被称为自动执行.

定义4 自动执行：区块链智能合约的自动执行是指部署在区块链上的链码在满足触发条件后履行合约或协议中条款的方式.

智能合约通常是应用软件的一部分，也是由

数字编码表示的条款代码, 尚不构成法律意义上的合同. 进而, 它可在不需要可信方的参与下由计算机网络执行, 且共识协议可确保执行正确性. 或者说, 智能合约是一种自动执行合约条款并自我验证和无需中介的计算机交易协议[10].

2　智能法律合约

2.1　智能法律合约定义及内涵

随着数字社会的快速发展以及区块链技术的广泛应用, 区块链智能合约的规范化需求日益强烈. 然而现有智能合约面临专业性强、可读性差、生产效率低等实际问题, 现实法律合同到可执行程序代码的高效转化难以实现, 这不仅影响了行业应用与计算机及法律界人士的跨领域合作, 而且阻碍了智能合约的法律化进程[11]. 有鉴于此, 智能法律合约被提出[12]:

定义 5 智能法律合约: 智能法律合约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.

智能法律合约本质上是一种符合法律的智能合约——“当事方之间以自然语言文本的形式, 在可计算逻辑的支持下, 以数字协议的形式在不同的计算机系统之间建立可移植的可执行义务”[13].

智能法律合约是一种介于现实法律合同与智能合约之间的过渡性手段. 如图 1 所示, 现实法律合同以自然语言为载体, 可翻译成由智能法律合约语言撰写的智能法律合约, 进而转化为由智能合约语言编写的智能合约.

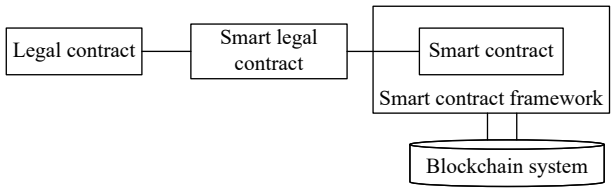

图 1　合同转化关系示意图

Fig.1　Diagram of contract transformation relationship

依据智能法律合约标准[12−13], 智能法律合约需遵循符合现行法律的合约结构及语法规则, 使智能法律合约满足以下性质:

(1)自然语言描述. 作为在智能合约和现实合约之间建立共同意思表示的桥梁, 智能法律合约必须是一种简单易懂、条理清晰的语言描述形式, 既兼有现实合约的法律特征和易理解性, 也具有智能合约的规范性和逻辑性.

(2)平台独立性. 智能法律合约所采用的语言属于解释型语言, 不直接产生目标机器代码, 而是通过解释器逐一将源程序语句解释成可执行的机器指令. 解释器中间层屏蔽了平台间的差异, 同样的智能法律合约在不同平台的执行结果是相同的, 使智能法律合约能够具有“一次编写, 随处运行”的平台独立性.

(3)可移植性. 作为现实合约和智能合约中间层次的合约形式, 智能法律合约的编写和解释器都应该对硬件或操作系统没有依赖性, 在不同的系统或平台运行时经过很少改动或不经修改就可解释编译成等效的智能合约程序, 即具有可移植性.

智能法律合约应建立在智能合约和现实合约之间, 既具有合同的法律特征和易理解性, 又有计算机程序代码的规范性[14−15], 以促进计算机、法律等专业人员的跨领域协作. 在区块链可确权基础上, 智能法律合约将物理世界有价值的资产, 包括房子、汽车、健康数据和版权等数字化为数字资产, 并与可编程数字法币相结合, 使其在区块链网络上自由使用和流通, 推动区块链智能合约快速健康发展. 同时, 智能法律合约以程序代码表达合同条款, 将现实法律合同与网络空间的程序代码相衔接, 可有效降低现实社会中法律合同生成智能合约的开发成本.

2.2　智能合约语言发展现状

编程语言是程序设计的基础, 理解智能合约语言也是掌握智能合约应用设计和运行机理的有效途径. 智能合约语言是利用智能合约实现现实业务的工具, 是帮助智能合约使用者撰写智能合约程序与代码的编程语言[16]. 从智能合约语言的语言特点和运行环境分析, 可将目前的智能合约语言分为三类:

(1)专有型智能合约: 是一种针对特定功能区块链系统而开发的特定智能合约. 典型代表是比特币脚本系统下的货币型合约, 它采用简单脚本指令系统和类 Forth 语言的栈式结构实现计算与条件控制.

(2)通用型智能合约: 使用常见程序设计语言, 运行在容器(Docker)与虚拟机(VM)中, 采用预定义接口与区块链进行交互. 例如, 超级账本[17] 支持如 Go、Java 等语言直接编写.

(3)专业型智能合约: 采用领域专用语言(DSL)针对特定专业领域而开发的智能合约, 如法律、保险、知识产权、供应链金融、银行清结算、企业贷款等领域, 这种智能合约的特点是专业性强但通用性差, 通常需要转化为通用型或专有型智能合

约才能实际进行应用.

专有型和通用型智能合约的撰写仍需较强的计算机专业基础, 对于其他专业人员有较高的壁垒, 因此, 就智能合约不同行业应用开发而言, 专业型智能合约将成为智能合约发展的主要趋势.

2.3　智能法律合约与法律合同的关系和区别

法律合同是双方或多方以一定的权利义务关系声明约束他们之间民事法律关系的协议, 且可由法庭强制执行. 在本文中认为符合法律要求、且满足可读性、具备合约必需组件的合约均可认定为法律合同, 是一个更为广泛的概念. 其中, 自然语言撰写的传统纸质合约是法律合同的最常见形式.

智能合约是否属于法律合同是学者一直争论的问题, 部分学者[18]认为智能合约通过代码表达当事人合意, 且能在部署过程中满足“要约—承诺”结构, 应属于法律合同; 但更多学者[19−20]认为它的信赖前提是智能合约技术及其自治秩序, 其代码在有效语义一致性识别时存在解释困难, 且一旦部署后的智能合约“不可撤回”, 将事实出现强制缔约的现象, 应属于基于自治联合体的多维信赖合同, 而不能认为其完全符合法律合同的要求, 本文对该问题持有相同态度.

智能法律合约是在传统现实合约基础上进行形式化提取、着重于合约程序性和法律性共举的合约形式, 可以通过转译机制转化成智能合约, 从而进行程序运行. 智能法律合约语言在规避自然语言二义性的同时引入其法律特征和程序语言的规范性, 所编写的智能法律合约可以更明确地表达合约条款、权利义务约束、权属交换等内容, 在合约内容中符合法律合同规定. 在本文后续论述中可以得知智能法律合约也具有和传统现实合约一样的法律效力, 且满足合同所需要件, 因此智能法律合约属于法律合同.

推论 1: 智能法律合约是对传统现实合约的形式化、模板化表达形式, 属于法律合同且可转化为智能合约.

3　智能合约的法律化探索与实践

3.1　智能合约的法律化思考

依照《中华人民共和国民法典》(简称《民法典》)第 464 条: “合同是民事主体之间设立、变更、终止民事法律关系的协议. ”这与定义 1 中智能合约概念相一致, 智能合约形式下当事人约定了多方应该遵循的行为约束, 签订合约的目的、性质并未发生变化[18], 但智能合约要成为意思自治的合理表现形式, 其形式合法化、内容规范化, 编写和运行过程都应该满足国家现有法律和政策框架的约束, 目前智能合约距离满足法律要求仍存在很多法律问题.

辨析 1: 传统合同采用自然语言、法言法语、专业术语与在智能合约中的计算机代码之间存在差异.

传统合同与智能合约的差异表现为:

① 传统合同为了针对各种无法预见的情况, 不但经常使用一些抽象的、概括的、灵活的语言以实现内容高度的通用性, 还经常大量使用法言法语甚至专业领域的术语;

② 智能合约采用了非二义性、形式化的计算机语言进行表达, 是一种可执行性的指令序列[21], 常使用严谨、正式、“死板”的语言将合约内容中的条件、范围等进行限定.

因此, 两种语言之间差异体现在两个方面: 二义性与确定性、抽象与具体, 相对而言, 智能合约语言误解的几率比传统合同更低, 这也是其优势所在.

另一方面, 智能合约作为计算机程序, 缺少对当事人权利和义务等法律关系与行为的明确表述[22], 难以由法律人士通过程序代码区分合约表述的权利义务关系, 这一缺点直接影响到智能合约法律效力, 也是必须克服的缺点.

其次, 依照《民法典》第 470 条规定: “合同的内容由当事人约定, 一般包括下列条款: (一)当事人的姓名或者名称和住所; (二)标的; (三)数量; (四)质量; (五)价款或者报酬; (六)履行期限、地点和方式; (七)违约责任; (八)解决争议的方法. ”智能合约作为一种电子合同, 其内容也应遵循上述规定的法律要素.

推论 2 文法要求: 智能合约法律化应遵循自然语言表述为基础、法言法语为标准词汇、计算机形式化表达为语法、法律要素为框架生成智能合约.

传统合同与智能合约之间在用语方面存在很大不同, 因此在两者之间的转化过程中也必然会出现问题而带来法律风险. 在产生纠纷需要法院或者仲裁机构进行裁判, 理解代码含义或代码逆向转化回合同条款时, 容易出现歧义或者模糊的用语(代码)难以界定, 使得法院或者仲裁机构难以作出裁判, 为诉讼带来更大的时间成本与经济成本.

辨析 2: 智能合约自动执行与法律合同中当事人享有权利之间存在的内生矛盾.

自动执行能力是智能合约最为鲜明的特征, 它在履行合同条款时具有无偏差执行、自动验证、不依赖其他机构等优点, 同时也具有履行条款中“机械性”的缺点, 即一旦条件满足被触发后将无法停止执行, 也不会被单方终止.

在讨论智能合约法律化时, 计算机不能承担法律责任是一个永恒的前提[23], 智能合约应保证其行为必须得到人的授权. 因此, 智能合约所具有的自动执行能力与法律合同中的当事人享有权利之间的内生矛盾体现如下:

① 智能合约的自动执行能力不能代替当事人的意志选择, 应允许当事人自行确定是否行使权利或履行义务, 以及权利义务履行的方式.

② 智能合约的自动执行能力不能通过禁止条款限制现实世界中人类的行为.

③ 智能合约应允许当事人因不可抗力、履约过程中外部条件发生变化或遇有特殊情况时, 经当事人协商一致后停止履行合同, 限制智能合约的自动执行能力.

智能合约有能力帮助当事人自动处理各种线上交易, 如自动转账等, 但在撰写智能合约时, 应该有意限制其不可代替当事人自动执行这些义务性行为(即智能合约不应具备自主权和独立权), 或者提供当事人对权利义务的选择. 因此, 智能合约自动执行的前提基础应当是: 只有当事人明确已授权情况下, 智能合约才可代替其执行指定的行为.

智能合约法律化的发展方向应该是对合约履行过程中当事人行为的结果进行检测和验证, 从而判断当事人在规定条款下是否完成意定的行为或者实现某种结果, 进而为合约履行提供便利.

推论 3 非赋权原则: 智能合约法律化过程中应注意约束智能合约不可代替当事人自动执行意志选择, 而应针对当事人行为结果进行条款履行的检测和验证.

根据《民法典》第 180 条规定: “因不可抗力不能履行民事义务的, 不承担民事责任. 法律另有规定的, 依照其规定. ”完备的智能合约系统应该支持不可抗力或特殊情况下, 终止履行合同乃至解除合同.

辨析 3: 经转化后的智能合约存在法律有效性与代码安全性问题.

与常规计算机程序一样, (经转化后)智能合约代码通常包括两部分: 表征合同内容的针对性代码与为了重复使用而预先拟定的引用性代码(包括各种类库、平台 Jar 包和函数库). 由于智能合约是由专业受限的程序员所撰写, 他们可能对合同条款缺乏审查能力, 会将本应无效的条款转化为代码或无法意识到合约是否存在违反法律要求的行为. 因此, 有必要对智能合约代码进行法律有效性审查, 包括:

首先, 对于针对性代码, 应该根据《民法典》中对无效民事法律行为(如欺诈、胁迫、虚假意思表示)的认定进行审查, 从而判断其效力作用. 此外, 传统合约转化至计算机代码目前尚缺少标准化的转化方式, 转化结果因人而异、参差不齐, 这些无疑增大了智能合约作为合约在法律认定上的难度.

其次, 对于引用性代码, 易于将其归属于法律合同中的格式条款. 依据《民法典》第 496 条: “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. 采用格式条款订立合同的, 提供格式条款的一方应当遵循公平原则确定当事人之间的权利和义务, 并采取合理的方式提示对方注意免除或者减轻其责任等与对方有重大利害关系的条款, 按照对方的要求, 对该条款予以说明. 提供格式条款的一方未履行提示或者说明义务, 致使对方没有注意或者理解与其有重大利害关系的条款的, 对方可以主张该条款不成为合同的内容. ”需对引用性代码进行当事人告知和共识; 同时, 需对该代码进行无效性审查, 即审查其是否违反《民法典》第 497 条规定: “有下列情形之一的, 该格式条款无效: (一)具有本法第一编第六章第三节和本法第五百零六条规定的无效情形; (二)提供格式条款一方不合理地免除或者减轻其责任、加重对方责任、限制对方主要权利; (三)提供格式条款一方排除对方主要权利. ”

再者, 对于恶意编程人员编写的智能合约, 由于合约当事人缺乏对代码的基本了解, 也可能无法察觉到合约是否存在漏洞或意思表示差异, 因此恶意编程人员可利用己方优势进行欺诈、胁迫等违法行为, 而当事人则可能落入“陷阱”且缺乏能为自己原本意思表示做支撑的证据, 在这种情况下, 智能合约自动履行后往往会产生争议且难以进行认定. 有鉴于此, 对智能合约代码进行安全性检测是非常必要的.

推论 4 审查准则: 智能合约法律化过程中应保证智能合约代码可进行有效性审查和安全性检测.

由此可见，只有智能合约满足上述 3 个基本规则，包括：（一）文法要求、（二）非赋权原则、（三）审查准则，才能使智能合约具有法律化特征.

3.2　智能合约的法律化探索

智能法律合约作为现实法律合同与智能合约之间的过渡性手段，是智能合约法律化的一个重要途径. 为了满足智能合约的法律化 3 个基本原则，采用领域专用语言（Domain-specific language, DSL）开发智能法律合约语言（Smart legal contract language, SLCL）是一条有效途径. 其中 DSL 是指专注于某个应用程序领域的计算机语言. 采用这种领域性语言实现合同意思表示，既能让当事人之间便于沟通，又可让出现违约问题时介入的第三方能够准确理解合同内容，使所签订合同具有公知的法律效力，促进智能合约和传统合约之间的语言差异问题的解决.

基于这种思想，近年来一种被称为高级智能合约语言已引起学术界的广泛关注. 2018 年 He 等[24]基于面向领域语言的思想提出了一种面向现实合约的智能合约描述语言（SPESC），该语言在智能合约和传统合约之间引入了智能法律合约的概念，明确定义了当事人的义务和权利、以及加密货币的交易规则，用固定格式的语法结构提供了面向非计算机专业人员的智能法律合约撰写方式，并可以根据该语法结构将撰写完成的智能法律合约编译成可执行的智能合约程序，为实现智能合约的法律化提供了一个很好的解决途径.

使用 SPESC 语言撰写的智能法律合约包含合约框架、合约名称、当事人描述、标的、资产表达式、资产操作、合约条款、附加信息和合约订立，其中，资产操作包括存入、取出、转移，合约条款包括一般条款、违约条款以及仲裁条款，其具体语法定义如表 1 所示，以 SPESC 语言编写的竞买合约如图 2 所示. SPESC 模块中对法律规定的合同内容均进行了相应显式定义，符合《民法典》对合同内容的要求.

基于上述语法结构，该文后续工作[25]中对 SPESC 系统性使用进行了完善，文中提出了高级智能合约层、智能合约层和机器代码执行层的 3 层智能合约系统框架，并给出 SPESC 到目标程序语言（以 solidity[26]为例）的转化规则. 该转换规则给出了 3 部分转化内容：

（1）根据当事人描述 Parties 生成目标语言合约中的当事人合约（群体当事人合约或个体当事人合约），用于管理当事人，记录关键事件，并提供相关统计和查询等功能.

（2）根据合约其他内容生成目标语言合约中的主体合约，包括两部分：合约属性、当事人的定义与初始化；条款的处理.

（3）目标语言合约中补充生成当事人人员管理，程序时序控制及异常检测等机制.

转化规则的制定以及对 SPESC 中表达式的进一步细化使智能法律合约能够在一定规则体系下进行转化，转化后的智能合约具有规范和统一的函数结构和逻辑表达，有效避免了不同程序员撰写智能合约带来的个人色彩和不确定性，也可确保转化后的智能合约与智能法律合约具有相同的意思表示，并可随时调取查看合约的运行信息.

2020 年，Wöhrer 和 Zdun[27]提出了基于 DSL 的面向合同的合约语言 CML，该语言在表现形式上类似于文献 [24] 的 SPESC 语言，使用状态变量和一组动作描述合同的规定动作和涉及的变量；使用“may”或者“must”作为条款的情态动词；使用关键字“due”表示时序关系，后跟时态优先级（即“after”或“before”）和触发器表达式，作者在文中也给出了从 CML 到 Solidity 的各部分转化方法.

总之，基于 DSL 构造的智能法律合约语言 SLCL 便于计算机、法律等各界人士理解，打破了之前所述传统合同和智能合约之间用语壁垒. 同时，所给出的转化规则将该语言所撰写的高级智能合约映射到目前可执行的智能合约，使得两者的“意思表示一致”.

根据我国《民法典》第 466 条规定：“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. ”法律所允许并承认：采用多种文字订立合同并约定具有同等效力，成立之后的合同皆是受法律保护的合同，对合同当事人具有法律拘束力. 因此，从现实合约到智能法律合约、智能合约、运行链码这一转化过程，尽管各阶段采用的语言不同，但用以表述相同含义的智能法律合约、智能合约、运行链码皆是具有相同法律效力的书面合同. 因此，我们有以下推论：

推论 5：智能合约及其链码与原合同文本具有同等法律效力，只要（一）采用智能法律合约生成与订立；（二）满足智能合约法律化三个基本规则；（三）约定具有同等效力.

需要注意的是，智能合约作为一种新的电子合同，是一个不断发展的概念，法律化进程也需要不

表 1　SPESC 语法模型

Table 1　SPESC grammar model

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+})

断完善, 如合约标的表示和实现、违约条款后续处理、订立过程规范化、证据保全、仲裁流程等内容.

4　智能合约系统架构及法律化辨析

4.1　智能合约架构描述

智能合约平台是智能合约架构的基础, 也为合约条款履行提供计算环境, 以区块链为基础, 智能合约平台通过提供丰富的编程语言, 使开发人员得以实现任意价值的交换[28]. 智能合约平台[29]定义如下:

定义 6(智能合约平台): 智能合约平台是指一种支持智能合约可执行程序部署、运行、验证的信息网络系统.

从计算模型角度来看, 区块链中的交易记录能够记录账户所持有货币的所有权状态以及预先定义好的"状态转换函数". 当其他交易或可信外部事件发生时, 它将依据"状态转换函数"转变为新的状态, 写入到交易记录并往复上述过程. 上述"状态转换函数"可视为预定义的合约代码, 表征当事人可行使权利和履行义务. 同时, 从法律视角来看, 激活"状态转移函数"的条款运行机制必须获得当事人授权.

```
contract SimpleAuction{

    party group bidders{
        amount : Money
        Bid()
        WithdrawOverbidMoney()
    }

    party auctioneer{
        StartBidding(reservePrice : Money, biddingTime : Date)
        StopBidding()
    }

    highestPrice : Money
    highestBidder : bidders
    BiddingStopTime : Date

    term no1 : auctioneer can StartBidding,
        when before auctioneer did StartBidding
        where highestPrice = reservePrice and BiddingStopTime = biddingTime + now.

    term no2 : bidders can Bid,
        when after auctioneer did StartBidding and before BiddingStopTime
        while deposit $ value > highestPrice
        where highestPrice = value and highestBidder = this bidder and
            this bidder::amount = this bidder::Ori amount + value .

    term no3_1 : bidders can WithdrawOverbidMoney,
        when this bidder isn't highestBidder and this bidder::amount > 0
        while withdraw $this bidder::amount
        where this bidder::amount = 0.

    term no3_2 : bidders can WithdrawOverbidMoney,
        when this bidder is highestBidder and this bidder::amount > highestPrice
        while withdraw $this bidder::amount - highestPrice
        where this bidder::amount = highestPrice.

    term no4 : auctioneer can StopBidding,
        when after BiddingStopTime and before auctioneer did StopBidding
        while withdraw $highestPrice.
}
```

图 2　SPESC 编写的竞买合约

Fig.2　Bidding contract written in SPESC

图 3 描述了区块链智能合约的通用架构. 该架构涉及到智能合约的程序设计、代码生成、部署与执行等多个阶段.

4.2　智能合约的区块链部署

智能合约的存储与执行、结果有效性、合约代码安全都依赖于区块链[30], 区块链与智能合约的有效整合成为智能合约实施的关键[31]. 为便于理解, 智能合约通常将区块链视为参与者共同维护的交易数据库; 智能合约将以交易形式被部署到区块链, 且可追加交易用于更改共享数据库中某些内容, 但该更改行为必须被其它所有参与方所接受, 这被称为“all-or-nothing”原则; 此外, 签名机制可用来保证只有持有该帐户密钥的人才能从该帐户中转移资金; 区块可视为时间上具有线性关系的存储单元, 而建立在区块之间的单向哈希链可视为因解决合约冲突而建立的公认交易顺序.

通过在智能合约中引入面向对象和面向领域的编程思想, 使得区块链中的各种复杂机制(挖矿、共识、对等网络、哈希等)变成了智能合约平台提供的承诺, 开发和使用人员只需要关注于自己的业务需求, 而无需考虑智能合约执行代码的具体实施.

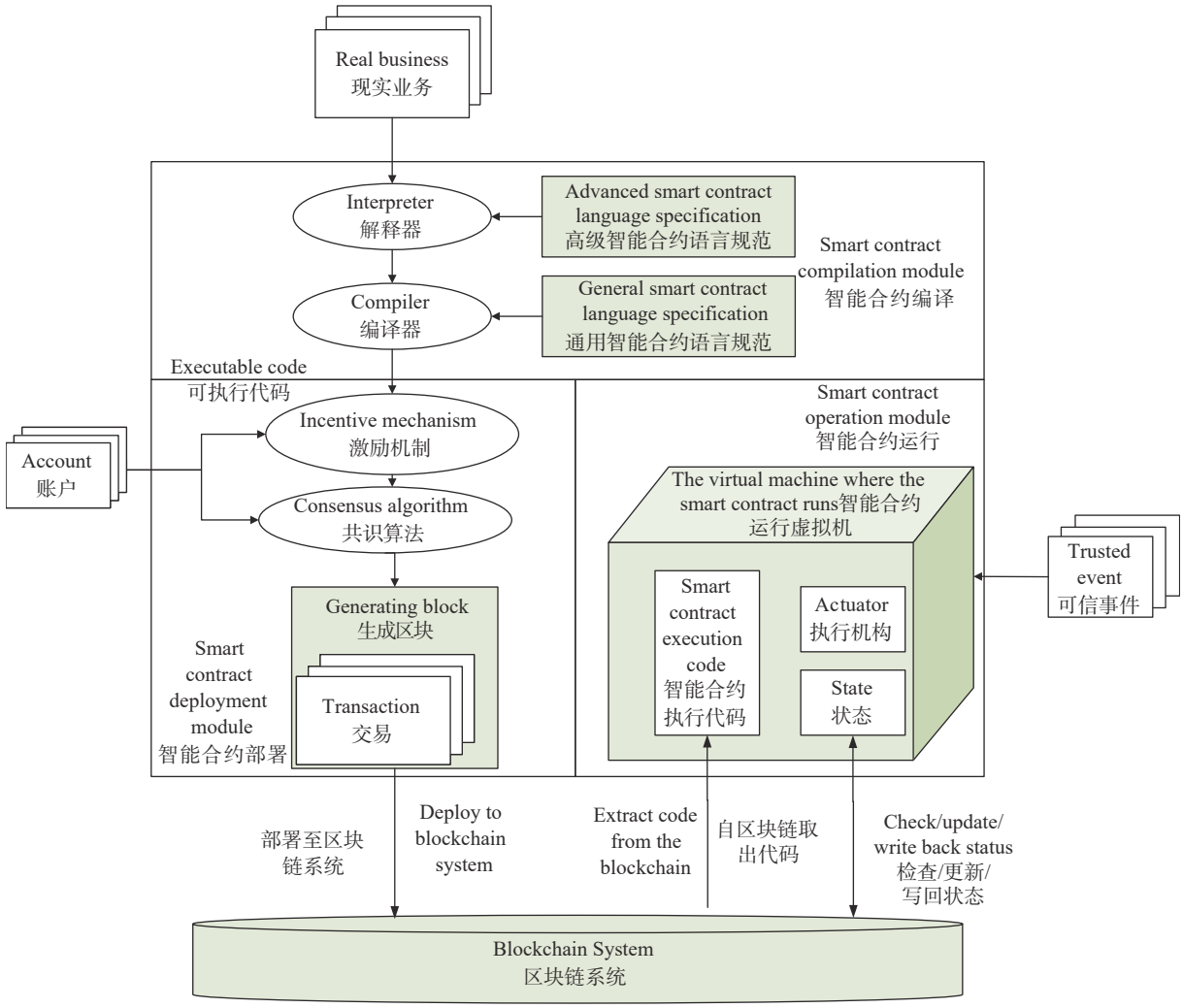

图 3　区块链智能合约通用架构

Fig.3　General architecture of blockchain smart contract

4.3　合约代码运行

当满足触发条件时, 被部署在区块链上的智能合约代码将被区块链系统自动执行, 并依照合约规定完成各种资产的转移. 为保证合约代码自动和无差错地被执行, 需要引入下列机制:

① 奖励机制是针对合约代码执行中的各种开销, 由发布者预付一定量的货币(如以太坊中以 gas 为单位的交易费用)作为奖励, 通常它是智能合约平台的必备条件.

② 执行机构是指合约代码运行的环境, 包括脚本、容器、虚拟机等 3 种运行方式[32], 此方面技术比较成熟, 是智能合约平台构造的核心技术.

③ 指令系统是智能合约运行环境提供的全部指令的集合, 反映了运行环境所拥有的基本功能[33], 是智能合约平台软件构建的基础.

④ 预定义的合约条款触发场景和响应规则是自动判定当前场景满足合约条款触发条件的依据, 响应规则则验证智能合约代码运行后的结果, 并向区块链发送用以更新合约状态的交易.

4.4　区块链所部署智能合约法律化辨析

联合国国际贸易法委员会《电子商务示范法》中将电子化的意思表示称之为“数据电文”. 根据我国《电子签名法》规定, 数据电文被定义为:

定义 7 数据电文: 数据电文是指以电子、光学、磁或者类似手段生成、发送、接收或者储存的信息.

区块链智能合约及其链码都是一种采用电子方式生成的计算机代码, 它被发送到区块链网络中, 并被网络中所有节点接收和存储, 因此根据上述定义, 不难证明下面辨析成立:

辨析 4: 区块链智能合约及其所生成的链码属于数据电文.

区块链智能合约是一种基于计算机语言、以电子方式生成的计算机代码, 链码则是通过智能合约语言编译器生成的可执行指令序列, 而且, 它们通过计算机网络被发送到区块链中每一个(完全)节点并在共识机制处理下被存储, 这些活动符合

数据电文中关于电子手段进行生成、发送、接收及存储的要求. 因此, 它们都符合数据电文规定.

区块链智能合约不仅是一种数据电文而且依据我国现行法律属于书面形式:

辨析 5: 区块链智能合约及其所生成的链码属于书面形式.

依据当前我国《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. " 部署于区块链智能合约平台上的合约代码应属于数据电文. 区块链智能合约及其所生成的链码虽然在形式上采用了计算机语言的指令代码形式进行表示, 对普通人员来说已经不具有可理解性, 但可以通过电子数据交换形式从区块链上随时调取查看上述相关代码, 并可通过屏幕显示或打印形式有形地表现所载内容, 因此它们属于书面形式.

可知, 区块链智能合约及其链码在形式上已经具有法律合同的特征, 符合当事人订立合同的要求. 然而, 在内容上区块链智能合约及其链码仍然无法满足我国现行法律的规定.

辨析 6: 以可执行指令序列表示的区块链智能合约及其所生成的链码不足以构成法律合同.

依照《民法典》第 470 条规定, 以可执行指令序列表示的区块链智能合约及其所生成的链码表述了计算机自动化执行的过程, 并不能以直观的方式确定上述法律合同要件.

下面以反例进行说明. 合同条款在法律上是合同条件的表现和固定化, 也是确定合约当事人权利和义务的根据. 然而, 结构化语言表示的智能合约通过以函数或过程形式描述当事人行为, 不支持能愿动词的使用, 因而无法直观描述执行人执行该行为的权利与义务关系, 有可能引起法律纠纷; 法律合同通过条款形式对当事人行为结果进行约定, 但某些条款并不直接限制达到该结果当事人的执行方式, 这与智能合约程序中确定性的某种执行指令序列并不一致. 因此, 以可执行指令序列表示的区块链智能合约及其所生成的链码本身不构成合同.

推论 6: 在合同订立过程中必须明确约定现实合约、智能法律合约、代码、链码的相互转化关系, 并对它们的意思表示进行说明.

本文中智能法律合约转化关系及相关要素间关系图如图 4 所示. 依据《民法典》第 143 规定: "具备下列条件的民事法律行为有效: (一)行为人具有相应的民事行为能力; (二)意思表示真实; (三)不违反法律、行政法规的强制性规定, 不违背公序良俗. "合同存在法律效力应该满足 3 方面的原则, 即: 1)行为人具有相应的民事行为能力; 2)意思表示真实; 3)不违反法律、行政法规的强制性规定, 不违背公序良俗.

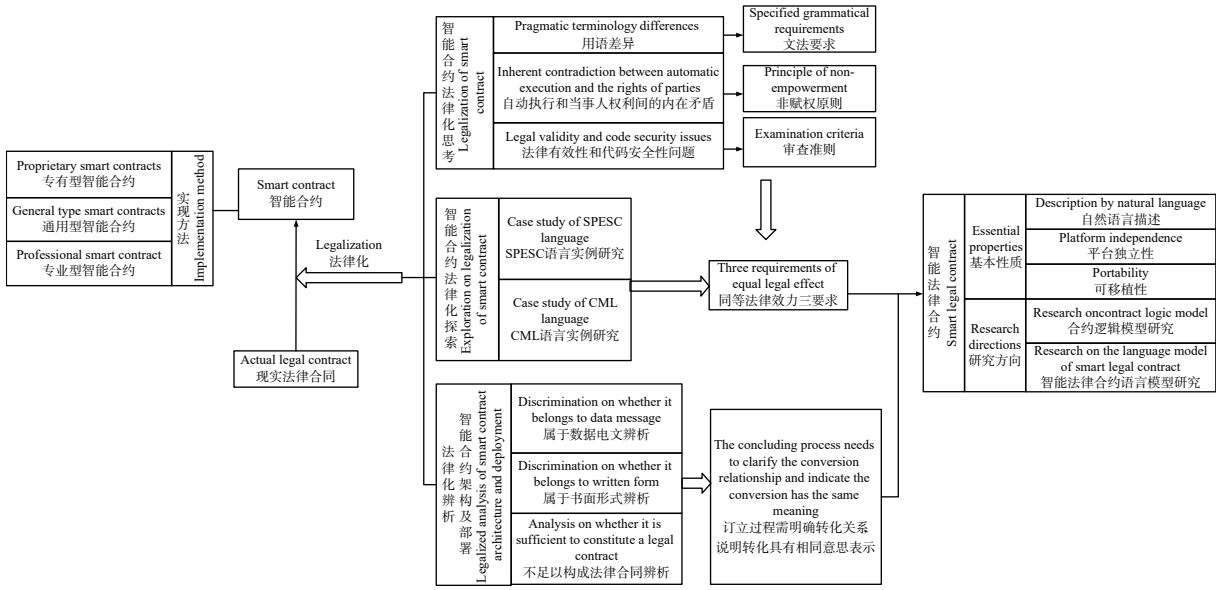

图 4　本文中智能法律合约转化关系及相关要素间关系图

Fig.4　Diagram of the transformation relationship between smart legal contracts

针对原则 1), 现实合约、智能法律合约、代码、链码的运行需经过所有当事人的签订, 且部分行为是需要当事人触发执行的, 最大限度保证了当事人具备民事行为能力; 针对原则 3), 现实合约

中若包含违反上述规定的行为, 现实合约将自动失效, 与此对应的智能法律合约、代码、链码也将自动视为无效合同, 从而可以保证满足原则 3)的要求.

针对原则 2), 现实合约具有法律约束力, 其中规定了全部的合同条款和各方的权利和义务, 而代码和链码皆具有极强的专业性和领域性, 在涉及其意思解读时会带来第三方理解的误差或困难, 难以验证.

综上, 在智能法律合约充当现实合约和智能合约(代码)、链码之间桥梁的基础上, 其意思表示一致是智能合约具有法律效力的必要条件, 因此在合同订立过程中, 必须保证它们间相互转化关系是可经过检验求证的, 且当事人签订时需知晓且同意其互证关系.

5　智能法律合约研究进展

智能法律合约虽然是一个较新的词汇, 但其涉及到的法律合约逻辑、智能合约形式化表达等基础探索和相关研究在很久前便已经开始了, 现行研究工作主要分为两个方向: 合约的逻辑模型研究与智能法律合约语言模型研究. 下面分别就这两方面研究进展予以介绍与分析.

5.1　合约逻辑模型研究

在早期论文中, 1952 年, von Wright 发表论文 Denotic logic[34], 被视为现代道义逻辑的开端, 他将义务、允许和禁止的规范概念与量化词——所有、某些、否, 以及模态词——必要、可能、不可能联系起来, 基于经典命题逻辑形成了标准道义逻辑(SDL)的基础, 可谓是形式化研究的开端. 1988 年, Lee[35] 对道义逻辑(Deontic logic)给出了更加详细的描述和扩充, 提出了一种强调合约时序性、道义性和行为性的逻辑合约模型, 成为最突出的合同正式化范式之一.

在此基础上, 2001 年 Grosof 等[36] 使用谦虚逻辑编程(Courteous logic program, CLP)给出了有关知识表示的新的形式化表示, 通过增加冲突的优先处理, 扩展自说明性的普通逻辑, 并开发了一个可以翻译任何 CLP 程序到语义相等的 OLP 程序的编译器.

Governatori 等在连续多年的研究中对道义逻辑表示合同的应用给出了更多的可行方案. 作者在论文 [37] 中利用谦虚逻辑编程开发了使用 RuleML(Rule markup language)语言表达的商务合同的推理机, 以机器可读语言来明确合同的所有条件, 将其转换为可执行代码, 并提供了一种将业务规则表示为模块化独立单元的方法, 兼有使用优先级和覆盖谓词来解决冲突的能力. 在论文 [38] 进一步研究了使用道义逻辑来表达合约语义的需要, 并提出了使用可废止逻辑对模态逻辑算子和道义逻辑算子的进一步扩展. 在论文 [39] 中提出了一个用来表示和推理电子合约的架构(DR-CONTRACT). 在道义逻辑的基础上增加了可废止逻辑, 使用道义可废止逻辑(Deontic defeasible logic)来处理合同违约情况.

2015 年, Idelberger 等[40] 提出基于逻辑的智能合约语言以替代过程式语言(Procedural language). 当合约中执行复杂逻辑时, 用过程式语言描述合约会使程序更加笨重且易错, 同时因为逻辑约束的动作执行顺序直接影响程序状态, 导致合约不易维护和监测. 而基于逻辑的语言可以避免这个问题, 其规则的顺序不影响程序执行, 并且语句声明将会更加简洁.

2016 年 Frantz 和 Nowostawski[41] 基于捕捉法律基本特征的制度语法(ADICO)[42] 提出了将人机可读的合同半自动翻译成智能合约的方法, 支持将 ADICO 组件映射为相应的 Solidity 结构, 包括生成合约地址、修饰器名称、函数框架, 但是尚未实现函数主体和 Solidity 表达式的转换.

2019 年王璞巍等[43] 以承诺(Commitment)作为基本元素构建了面向合同的智能合约形式化语言, 承诺表示为C(x,y,p,r,tc), 代表承诺人 x 向被承诺人 y 做出承诺, 如果前提 p 达成, 就产生结果 r, tc 表示该承诺的有效期限, tc 为 true 时承诺才会有效. 上述工作表明合约逻辑模型日趋完善, 并正朝着与现实法律相融合的方向发展.

5.2　智能法律合约语言模型研究

文献 [44] 提出了一种被称为 Simplicity 的功能性语言, 该语言通过对抽象机器上操作语义的评估, 计算空间和时间资源消耗的上限, 在执行之前计算出比特币脚本和以太坊虚拟机中的资源消耗费用, 有利于解决智能合约的预付费问题.

文献 [45] 提出了一种声明式智能合约语言(Findel), 着重从金融的角度描述了合约的资金转移动作及乘法、逻辑、时序表达式. Findel 通过两种资金转移动作与乘法、逻辑、时序 3 类表达式的组合编写合约, 使用该语言可以将某借款合约体现为一个表达式, 记为 c_{zcb}. 其合约最终体现为一个表达式, 如下所示:

$$c_{\mathrm{zcb}}=\left\{\begin{array}{l}\mathrm{And}(\mathrm{Give}(\mathrm{Scale}(10,\mathrm{One}(\mathrm{USD}))),\\ \mathrm{At}(\mathrm{now}+1\mathrm{years},\mathrm{Scale}(11,\mathrm{One}(\mathrm{USD}))))\end{array}\right\}$$

该表达式表示：出借人向借款人借款 10 USD，一年后借款人还款 11 USD. 由此可见，Findel 可以表示具有时序关系的简单金融合约，但无法支持变量的定义，且一个合约只能涉及两个当事人，功能较为单一，表达能力有限.

文献 [46] 以可读性与安全性为目标提出了一种类自然智能合约语言（SmaCoNat）. 如图 5 所示，SmaCoNat 合约包含合约头、账户、资产、协议、事件. 其中，合约头、账户、事件分别类似于 SPESC 中的合约名称、当事人、条款，而资产与协议分别用来声明合约中涉及的资产以及对资产初始化. SmaCoNat 对于资产做出了更加具体的描述与限定，但依旧存在问题：①没有表达如何在合约中存储信息，只支持对于资产转移的描述，因此应用范围较小；②没有对于时序的控制，每个输入事件之间相互独立，仅通过资产进行联系，条款之间的关系更难梳理与理解；③无法转化为可执行的智能合约编程语言.

```
Contract in SmaCoNat version 0.1.

§ Involved Accounts:
Account 'BarrierIn' by 'AComp' by Genesis alias 'BarrierIn'.
Account 'BarrierOut' by 'AComp' by Genesis alias 'BarrierOut'.

§ Involved Assets:
Asset 'TheCoin' by Genesis alias 'TheCoin'.
Asset 'ParkTicket' by Self alias 'Ticket'.
Asset 'OpenBarrier' by Self alias 'Open'.

§ Agreement:
Self issues 'Ticket' with value 42.
Self issues 'Open' with value 1.

§ Input Event:
if Input is equal to 'TheCoin' from Anyone
and if value of Input is equal to 0.3
then
    Self transfers 'Ticket' with value 1 to owner of Input.
    Self transfers 'Open' with value 1 to 'BarrierIn'.
    Self issues 'Open' with value 1.
endif

if Input is equal to 'Ticket' from Anyone then
    Self transfers 'Open' with value 1 to 'BarrierOut'.
    Self issues 'Open' with value 1.
endif
```

图 5　SmaCoNat 合约

Fig.5　Contract written in SmaCoNat

文章 [47] 提供了一个新的自动生成智能合约的框架，其框架利用语义规则对特定领域的知识进行编码，然后利用抽象语法树的结构来合并所需的约束，最终通过经过约束的语法编码为区块链的智能合约. 其智能合约语言采用改进后的网络本体语言（OWL）——语义网规则语言（SWRL）描述智能合约. 尽管该文献提供了从 SWRL 自动转化为智能合约的生成器，但是该语言采用本体论语言的语法表示，不易读写，且主要应用于对数据的限制与检验，缺乏对于数据、合约状态变化以及金融方面的描述.

上述现有研究中多利用逻辑学的相关成果，针对某一领域方向，如金融领域展开研究，从而建立现实合约和智能合约之间的关联，形成智能法律合约. 总体来看，智能法律合约的相关研究普遍存在表达能力有限、触发控制机制不足、转化至可编程语言的能力较弱等问题.

6　总结与展望

智能合约的法律化，必然导致计算机程序开发新的变革. 针对智能合约如何有效应对法律化要求，学者们对此开展过讨论，形成了富有启发性的观点. 本文对智能法律合约的各种研究进展进行了归纳总结，阐述了智能合约的法律化探索和实践，对法律化进程中的关键问题进行了辨析与总结.

智能合约是一个新兴的领域，尤其是智能法律合约的研究应保持前瞻性，不能等待智能合约法律地位确定后再发展智能合约产业. 与其相比较，法律的滞后性是不可避免的，因为法律不能假设、不能预想可能发生什么，进而对该可能性进行立法. 因此，利用智能法律合约推进智能合约法律化具有积极意义.

参　考　文　献

[1] Nakamoto S, Bitcoin: a peer-to-peer electronic cash system [J/OL]. *Bitcoin Online* (2008-10-31) [2021-09-17] https://bitcoin.org/bitcoin.pdf

[2] Szabo N. Smart contracts. [J/OL] *Internet Documentation Online* (2018-05-30)[2021-09-29].https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinter-school2006/szabo.best.vwh.net/smart.contracts.html

[3] Szabo N. Smart contracts: building blocks for digital markets. *EXTROPY: J Transhumanist Thought*, 1996(16): 18

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

[5] Bertoli P. *Smart* (*legal*) *Contracts: Forum and Applicable Law Issues*. Berlin: Springer International Publishing, 2020

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

[7] Savelyev A. Contract law 2.0:'Smart' contracts as the beginning of

the end of classic contract law. *Inf Commun Technol Law*, 2017, 26(2): 116

[8] Fries M. *Smart Contracts*. Tübingen: Mohr Siebeck, 2019

[9] 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)

[10] Singh A, Parizi R M, Zhang Q, et al. Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities. *Comput Secur*, 2020, 88: 101654

[11] Wang S, Ouyang L W, Yuan Y, et al. Blockchain-enabled smart contracts: Architecture, applications, and future trends. *IEEE Trans Syst Man Cybern:Syst*, 2019, 49(11): 2266

[12] Chinese Institute of Electronics. T/CIE 159 —2020 *Formal Expression of Blockchain Smart Contract*. Beijing: Standards Press of China, 2021
(中国电子学会. T/CIE 159—2020区块链智能合约形式化表达. 北京: 中国标准出版社2021)

[13] Pereira J C. The genesis of the revolution in contract law: Smart legal contracts//*Proceedings of the 12th International Conference on Theory and Practice of Electronic Governance*. Melbourne, 2019: 374

[14] 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

[15] Gelati J, Rotolo A, Sartor G, et al. Normative autonomy and normative co-ordination: Declarative power, representation, and mandate. *Artif Intell Law*, 2004, 12(1-2): 53

[16] He X D, Yi J Z, Chen A B. Application progress and development trend of block chain technology. *World Sci Tech R D*, 2018, 40(6): 615
(何小东, 易积政, 陈爱斌. 区块链技术的应用进展与发展趋势. 世界科技研究与发展, 2018, 40(6): 615)

[17] Androulaki E, Barger A, Bortnikov V, et al. Hyperledger fabric: A distributed operating system for permissioned blockchains // *Proceedings of the Thirteenth EuroSys Conference*. Porto Portugal, 2018: 1

[18] Lang F. New interpretation of smart contracts based on blockchain technology from the contractual perspective. *J Chongqing Univ Soc Sci Ed*, http://kns.cnki.net/kcms/detail50./1023.C.20200402.1345.004.html
(郎芳. 区块链技术下智能合约之于合同的新诠释. 重庆大学学报(社会科学版)http://kns.cnki.net/kcms/detail/50.1023.C.20200402.1345.004.html)

[19] Wu Y. The civil legal status of smart contracts. *Jurist*, 2020(2): 1
(吴烨. 论智能合约的私法构造. 法学家, 2020(2): 1)

[20] 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)

[21] He H W, Yan A, Chen Z H. Survey of smart contract technology and application based on blockchain. *J Comput Res Dev*, 2018, 55(11): 2452
(贺海武, 延安, 陈泽华. 基于区块链的智能合约技术与应用综述. 计算机研究与发展, 2018, 55(11): 2452)

[22] Meng B, Liu Q, Wang D J, et al. Review on conformance between legal contract and smart contract. *Application Research of Computers*, https://doi.org/10.19734/j.issn.1001-3695.2019.12.0652
(孟博, 刘琴, 王德军, 等. 法律合约与智能合约一致性综述. 计算机应用研究, https://doi.org/10.19734/j.issn.1001-3695.2019.12.0652)

[23] Si X, Cao J F. On the civil liability of artificial intelligence. *Sci Law* (*J Northwest Univ Political Sci Law*), 2017, 35(5): 166
(司晓, 曹建峰. 论人工智能的民事责任: 以自动驾驶汽车和智能机器人为切入点. 法律科学(西北政法大学学报), 2017, 35(5): 166)

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

[25] Zhu Y, Qin B H, Chen E, et al. An advanced smart contract conversion and its design and implementation for auction contract. *Chin J Comput*, 2021, 44(3): 652
(朱岩, 秦博涵, 陈娥, 等. 一种高级智能合约转化方法及竞买合约设计与实现. 计算机学报, 2021, 44(3): 652)

[26] Ethereum. Solidity Documentation-Release 0.8. 8 [J/OL]. *Internet Documentation Online* (2021-9-15) [2021-09-16].https://buildmedia.readthedocs.org/media/pdf/solidity/develop/solidity.pdf

[27] Wöhrer M, Zdun U. Domain specific language for smart contract development // 2020 *IEEE International Conference on Blockchain and Cryptocurrency* (*ICBC*). Toronto, 2020: 1

[28] Christidis K, Devetsikiotis M. Blockchains and smart contracts for the Internet of Things. *IEEE Access*, 2016, 4: 2292

[29] Zhu Y, Gan G H, Deng D, et al. Security architecture and key technologies of blockchain. *J Inf Secur Res*, 2016, 2(12): 1090
(朱岩, 甘国华, 邓迪, 等. 区块链关键技术中的安全性研究. 信息安全研究, 2016, 2(12): 1090)

[30] Sklaroff J M. Smart contracts and the cost of inflexibility. *Univ Pe Law Rev*, 2017, 166(1): 263

[31] Ouyang L W, Wang S, Yuan Y, et al. Smart contracts: Architecture and research progresses. *Acta Autom Sin*, 2019, 45(3): 445
(欧阳丽炜, 王帅, 袁勇, 等. 智能合约: 架构及进展. 自动化学报, 2019, 45(3): 445)

[32] Fang Y, Cong L H, Yang Z B. Study on digital smart contract based on blockchian. *Comput Syst Appl*, 2019, 28(9): 225
(方轶, 丛林虎, 杨珍波. 基于区块链的数字化智能合约研究. 计算机系统应用, 2019, 28(9): 225)

[33] Beaumont P H. *Fixed-income Synthetic Assets: Packaging, Pricing, and Trading Strategies for Financial Professionals*. New Jersey: John Wiley & Sons, 1992

[34] Von Wright G H. Deontic logic. *Mind*, 1951, LX(237): 1

[35] Lee R M. A logic model for electronic contracting. *Decis Support Syst*, 1988, 4(1): 27

[36] Grosof B N, Labrou Y, Chan H Y. A declarative approach to business rules in contracts: courteous logic programs in XML // *Proceedings of the* 1*st ACM conference on Electronic commerce*. Colorado, 1999: 68

[37] Governatori G. Representing business contracts in RuleML. *Int J Coop Info Syst*, 2005, 14(2-3): 181

[38] Governatori G, Milosevic Z. A formal analysis of a business contract language. *Int J Coop Info Syst*, 2006, 15(4): 659

[39] Governatori G, Pham D H. Dr-contract: An architecture for e-contracts in defeasible logic. *Int J Bus Process Integr Manag*, 2009, 4(3): 187

[40] Idelberger F, Governatori G, Riveret R, et al. Evaluation of logic-based smart contracts for blockchain systems // *International Symposium on Rules and Rule Markup Languages for the Semantic Web*. Galway, 2016: 167

[41] Frantz C K, Nowostawski M. From institutions to code: Towards automated generation of smart contracts // 2016 *IEEE* 1*st International Workshops on Foundations and Applications of Self* Systems* (*FAS*W*). Augsburg, 2016: 210

[42] Mavridou A, Laszka A. Designing secure ethereum smart contracts: A finite state machine based approach // *Proceedings of International Conference on Financial Cryptography and Data Security*. Berlin, 2018: 523

[43] Wang P W, Yang H T, Meng J, et al. Formal definition for classical smart contracts and reference implementation. *J Softw*, 2019, 30(9): 2608
(王璞巍, 杨航天, 孟佶, 等. 面向合同的智能合约的形式化定义及参考实现. 软件学报, 2019, 30(9): 2608)

[44] O'Connor R. Simplicity: A new language for blockchains // *Proceedings of the* 2017 *Workshop on Programming Languages and Analysis for Security*. Dallas, 2017: 107

[45] Biryukov A, Khovratovich D, Tikhomirov S. Findel: secure derivative contracts for ethereum. *Financial Cryptogr Data Secur*, 2017: 453

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

[47] 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