基于时间自动机的数据流通控制建模及验证

李恒[1,2,3,4]，李凤华[1,2,3]，梁琬珩[1,5]，郭云川[1,2,3]，张玲翠[1,2,3]，周紫妍[1,2,3]
（1.中国科学院信息工程研究所，北京 100085；2.中国科学院大学网络空间安全学院，北京 100049；
3.网络空间安全防御全国重点实验室，北京 100085；4.自然资源部国家基础地理信息中心，北京 100830；
5.南京信息工程大学计算机学院、网络空间安全学院，江苏 南京 210044）

摘　要：为了解决数据跨域流通控制策略生成、传递与执行的可行性、正确性和安全性验证难题，提出了一种基于时间自动机和计算树时序逻辑的形式化建模及验证方法。该方法首先针对数据流通控制流程，以及数据交易场景（模式）下的数据提供者、数据使用者（含数据经纪人）和数据监管者等实体分别进行形式化建模；随后给出了数据交易过程中，安全需求性质和流通控制属性的计算树时序逻辑形式化规约描述；最后，对上述时间自动机模型进行仿真，并对其性质和属性进行形式化验证与分析。实例分析表明，所提方法可以有效验证数据流通控制机制的可行性、正确性和安全性。

关键词：数据要素流通；访问控制；时间自动机；延伸控制；形式化方法验证
中图分类号：TN92
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025038

Modeling and verification of data circulation control based on timed automata

LI Heng[1,2,3,4], LI Fenghua[1,2,3], LIANG Wanheng[1,5], GUO Yunchuan[1,2,3],
ZHANG Lingcui[1,2,3], ZHOU Ziyan[1,2,3]
1. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China
2. School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
3. State Key Laboratory of Cyberspace Security Defense, Beijing 100085, China
4. National Geomatics Center of China, Ministry of Natural Resources of the People's Republic of China, Beijing 100830, China
5. School of Computer Science, School of Cyber Science and Engineering, Nanjing University of Information Sciences & Technology, Nanjing 210044, China

Abstract: To address the challenges of verifying the feasibility, correctness, and security of cross-domain data circulation control policies in their generation, transmission, and execution, a formal modeling and verification method was proposed based on timed automata and computation tree logic (CTL). Firstly, the formal models were established for the data circulation control process and key entities in data transaction scene, including data providers, data consumers (encompassing data brokers), and data supervisors. Subsequently, the security requirements and circulation control properties were formalized during data transactions using CTL specifications. Finally, the aforementioned timed automaton model was simulated, with formal verification and analysis performed on its behavioral properties and structural attributes. The proposed method can effectively validate the feasibility, correctness, and security of data circulation control mechanisms.

Keywords: data elements circulation, access control, timed automata, extension control, formal method verification

收稿日期：2025–02–12；修回日期：2025–03–13
通信作者：郭云川，guoyunchuan@iie.ac.cn
基金项目：国家重点研发计划基金资助项目（No.2023YFB3106505）；国家自然科学基金资助项目（No.U24A20240，No.62441226）
Foundation Items：The National Key Research and Development Program of China (No.2023YFB3106505), The National Natural Science Foundation of China (No.U24A20240, No.62441226)

0 引言

数据要素合规高效流通是激活数据潜能、推动数字经济化发展的关键。自《中共中央 国务院关于构建数据基础制度更好发挥数据要素作用的意见》（“数据二十条”）发布以来，2024年10月，中共中央办公厅、国务院办公厅正式发布《关于加快公共数据资源开发利用的意见》，要求以促进公共数据合规高效流通使用为主线，充分释放公共数据要素潜能，推动高质量发展。国家数据局会同有关部门发布了《可信数据空间发展行动计划（2024—2028年）》《关于促进数据产业高质量发展的指导意见》《关于完善数据流通安全治理 更好促进数据要素市场化价值化的实施方案》《公共数据资源授权运营实施规范（试行）》等一系列政策文件，要求将安全贯穿数据供给、流通、使用全过程，充分利用可信数据空间、数据流通控制、数联网、隐私保护计算平台等技术，建立高效便利可信的数据流通机制，促进数据大规模、低成本、安全自由流通。

数据流通安全[1]重点关注机密性、完整性和访问控制，数据流通控制即采用交易契约关联的数据使用策略、数据使用策略与数据安全绑定、数据使用策略可信执行、策略执行远程验证等技术措施，解决域内非授权访问、域外使用不受控等问题，实现对数据在使用方的流通控制和远程验证，约束多轮交易后数据依照策略契约使用。目前，由于缺乏对数据跨域流通控制过程的抽象化研究，难以对数据跨“运维管理控制域”后的延伸控制策略生成、流通控制策略可信执行、策略执行远程验证的可行性、正确性和安全性进行评估及系统性分析。因此，亟须对数据流通控制机制进行形式化规约描述，分析其可行性、正确性和安全性，并基于时间自动机（TA, timed automata）模型对数据流通控制机制进行建模与形式化方法验证。

20世纪90年代，Alur等[2]最早提出了时间自动机模型，该模型在有限自动机模型基础上引入时钟变量，并提出了与之相关的时钟约束，讨论了时间自动机的语言和可计算性问题，提出了如何通过模型检查验证实时系统中的时间约束。21世纪以来，国内外学者在时间自动机基础上进行了系统建模、流程测试、策略验证、模型检测等深入研究和应用工作，以UPPAAL[3-4]为代表的可视化实时系统验证工具在工业界和学术界得到广泛应用。Chechik等[5]探索了时间自动机与时间弧Petri网的结合，提出了这两者在验证和应用中的交集。Katoen等[6]提出了如何使用模型检查技术验证时间自动机，并讨论了相关的算法和工具。文献[7-12]针对时间自动机的公理化和UPPAAL验证工具的研发开展了进一步的模型检测优化工作，并取得高水平研究成果。

近年来，国内外研究者结合符号模型检查、组合验证方法、物联网（IoT, Internet of things）、机器学习等新技术，展开了一系列新的热点研究工作。Chechik等[13]针对现代网络化的物理系统（CPS, cyber-physical system）的应用需求，提出了一种更高效的时间自动机验证方法。Bjørner等[14]探讨了时间自动机在嵌入式系统中的应用，特别是在具有定量性能要求（如功耗、响应时间等）的系统中进行模型检查的策略。Larsen等[15]扩展了传统的时间自动机模型，加入了概率元素，使其能够处理更加复杂的不确定性和随机性系统的验证。Pérez等[16]给出了UPPAAL验证工具在自动驾驶系统中的应用，尤其是在时间约束和安全性能验证方面。Wang等[17]和Sun等[18]提出了符号时间自动机的方法，能够在更高效的模型检查框架中处理复杂的嵌入式系统和物联网系统。Zhao等[19-20]分别将时间自动机与博弈论相结合，提出了多智能体系统中使用时间自动机进行时序决策分析的方法，同时探讨了UPPAAL验证工具与符号模型检查的结合，提出了该方法在网络和嵌入式系统中的应用，尤其是在处理大规模系统时的优势。Liu等[21]研究了在网络安全和数据传输时延的背景下，如何使用UPPAAL验证工具对IoT系统进行安全性和时序验证。上述研究均面向嵌入式、物联网、自动驾驶等实时系统模型验证，但针对数据流通控制时序系统的形式化建模和验证研究尚处于空白。

流通控制技术是当前数据要素安全流通的一种重要方法。现有流通控制机制通常缺少验证，策略的可信生成、策略本身的精确传递和策略在异地的执行过程往往具有一定的潜在风险。因此，本文针对数据跨域流通控制（包括延伸控制策略生成、策略可信执行、策略执行远程验证等）可行性、正确性和安全性验证难题，采用时间自动机进行建模并给出形式化方法定义，对数据跨运维管理域延伸控制、受控流转策略可信执行、销毁删除策略执行等

进行形式化方法验证。本文主要的研究工作如下。

1) 在描述数据流通控制时序逻辑的基础上，基于数据资源的通用属性和安全属性，分别对数据提供者、数据使用者（含数据经纪人）、数据监管者等实体和流通控制过程开展了形式化建模，并给出了数据交易场景（模式）下的时间自动机模型。

2) 在数据交易场景（模式）下，将已建立的实体模型通过交易协商、数据提供与销毁删除等控制通道组合成时间自动机模型，确定具体数据流通延伸控制策略，对执行时序逻辑公式进行组合与约减，并开展了安全需求性质和流通控制属性的计算树时序逻辑形式化规约描述与分析。

3) 在 UPPAAL 验证工具中，将上述模型和计算树时序逻辑代入模拟器中，建立验证策略执行结果与跟踪的关系，最后代入验证器中进行检查验证，得到验证结果并执行反例跟踪。仿真与验证结果表明，基于时间自动机和计算树时序逻辑的形式化建模及验证方法可以有效验证数据流通控制的可行性、正确性和安全性。

1　预备知识

1.1　时间自动机

时间自动机[8]是一种扩展了经典有限状态机的自动机模型，特别适用于处理涉及时间的系统。它结合了传统的状态机和时钟变量，以模型化系统中事件的时间约束。

定义 1　时钟约束 ϕ。设时钟变量有限集为 C，在 C 上定义时钟约束 ϕ，其语法定义为 $\phi = \text{true}|x\#c|\phi \wedge \phi$。其中，$x \in C$，$c \in \mathbb{N}$，$\# \in \{<, \leqslant, =, >, \geqslant\}$，$\Phi(C)$ 是定义在 C 上的所有时钟约束 ϕ 的集合。

时钟解释 v 是从 C 到 $\mathbb{R}^+ \cup \{0\}$ 的一个映射，即时钟赋值函数为 $v{:}C \to \mathbb{R}^+ \cup \{0\}$，为时钟变量有限集 C 中的每个时钟变量指派一个实数值。

时钟变量有限集 C 的子集为 $\mathbb{C}$，$\mathbb{C} = 0$ 表示对 $\mathbb{C}$ 的所有时钟变量 c 赋值为 0（即时钟复位），对 $C - \mathbb{C}$ 的时钟变量没有影响。

定义 2　时间自动机 A。设时间自动机为六元组 $<L,l_0,C,\Sigma,E,I>$，其中，L 是位置（状态）的有限集，$l_0 \in L$ 是初始位置（状态），C 是时钟变量有限集，Σ 是动作（事件）的有限集，E 是边（即状态转移）的集合，$E \subseteq L \times \text{Guard} \times \Sigma \times 2^C \times L$，$\text{Guard} \in \Phi(C)$，$I \in L$ 是具有时钟约束的位置（状态）映射。

规则转换 $<l,\Psi,\phi,\delta,l'>$ 表示当位置（状态）为 l 的时钟满足时钟约束 φ 时，则系统可以完成动作（事件）Ψ 从初始位置（状态）l 转移到位置（状态）l'，并完成 δ 中的时钟复位。

定义 3　时间自动机语义 $<Q,R>$。时间自动机 A 的语义是一个位置（状态）转换系统 $<Q,R>$，其中，$Q = L \times C$，$R \subseteq Q \times \Gamma \times Q$，其中 $\Gamma = \Sigma \cup \mathbb{R}^+$。位置（状态）转换系统的某个位置（状态）用 $<l,c>$ 表示，其中，l 表示 $<Q,R>$ 的某个位置（状态），c 表示当前的一个时钟值，通常包括时延转换和动作转换 2 种类型。

时延转换。对于位置（状态）$<l,c>$ 和时延增量 $\varepsilon \geqslant 0$，如果对 $\forall \varepsilon \geqslant \varepsilon' \geqslant 0$，$c + \varepsilon' \in I(l)$，那么 $<l,c> \to <l + \varepsilon>$。

动作转换。对于位置（状态）$<l,c>$ 和位置（状态）转移 $<l,\Psi,\phi,\delta,l'>$，有 $c \in \phi$，那么 $<l,c> \rightarrowtail <l',c[\delta = 0]>$。

1.2　计算树逻辑与线性时序逻辑

计算树逻辑（CTL, computation tree logic）和线性时序逻辑（LTL, linear temporal logic）是用于描述与验证时序逻辑的形式化语言[22-23]，特别适用于描述动态系统中状态随时间的变化。CTL*[24]通过结合 CTL 和 LTL 的路径量词（PQ, path quantifier）与状态量词（SQ, state quantifier），能够表达系统的行为演变和状态演变，如表 1 所示。

表 1　**CTL 和 LTL 公式及含义**

公式	含义
Ap	对于所有路径，系统总是满足 p
EFp	存在一条路径，最终会有一个状态满足 p
$A(p \to Xq)$	对于所有路径，如果当前位置（状态）满足 p，则下一个位置（状态）必须满足 q
Fp	在未来的某个时刻，p 为真
EGp	存在一条路径，在该路径上从某个时刻开始，系统的所有后续位置（状态）都满足 p
$A(pUq)$	对于所有路径，p 会一直持续，直到 q 变为真

定义 4　原子命题与逻辑连接词。描述系统位置（状态）的基本陈述 p，表示“系统处于某一特定位置（状态）”。逻辑连接词与 ∧、或 ∨、非 ¬、

蕴含→等，用于连接不同位置（状态）表达式。

定义5　路径量词。全称量词A，表示“对于所有可能的路径”；存在量词E，表示“对于存在的某些路径”。

定义6　位置（状态）量词。下一个位置（状态）量词X，表示“下一个位置（状态）”；最终位置（状态）量词F，表示“某个时刻未来位置（状态）为真”；全局位置（状态）量词G，表示“在所有未来的位置（状态）都为真”；直到位置（状态）量词U，表示“某个条件会在未来某个时刻为真，且在此之前另一个条件一直为真”。

2　数据流通控制技术

2.1　数据流通控制需求

数据流通[1]是指数据在不同主体之间传输的过程，可抽象为一种伴随着价值和安全信任传递的，数据资源在数据提供方和数据需求方之间的交换与转移过程。该过程涉及数据所有权、管理权、使用权、收益权等权益的转移。数据流通基本模型如图1所示，管理实体可分为提供者（通常作为数据提供方和权益方）、管理者（通常作为数据管理方和监管方）和使用者（通常作为数据需求方和消费方）3类。通常包括开放、共享、交易和交换等流通方式。

数据流通的合规高效要求使数据成为数据要素必须具有可控性。数据要素合规高效流通具有交易多轮动态、交易数据出域多次流转、流通控制自适应变更[25]等典型特征。因此，迫切需要解决数据流通过程中（包括访问、加工、删除、脱敏、流转管控、边界过滤、追踪溯源、违规判定、审计取证等操作）的可控性[26-28]问题，从而保障多方数据在流通过程中可管可控，有效解决了数据安全共享与融合增值的问题。因此，数据流通控制在使用控制（UCON, usage control）模型[29]基础上，采用技术手段对数据的传输、存储、使用和销毁环节进行延伸控制，将数据流通控制意愿转化为可机读处理的策略契约，以解决数据可管可控的前置性问题，实现对数据资产使用的时间、地点、主体、行为和客体等因素的全生命周期控制。

2.2　数据流通控制流程

数据流通控制流程包含数据全生命周期的受控流转、销毁确认和流通监管等诸多内容。其中，受控流转包括：1) 延伸控制。数据跨域流通不改变数据所有权和管理权，流通控制策略的形成需要满足符合数据使用权交易、限定数据接收方按策略契约使用、控制数据使用范围、约束二次传播、解决流通控制的远程验证等要求；2) 策略的协商与生成。数据跨域流通控制采用契约关联数据使用策略、数据使用策略与数据资源安全绑定等技术。因此，流通控制策略的形成应充分考虑数据使用者使用数据的条件、数据使用者的权限和义务等因素，确保数据要素跨域流通受控使用。

销毁确认包括：1) 多副本完备删除。在数据资源跨域流通过程中形成衍生数据，当契约到期时，数据流通控制策略应执行源数据和衍生数据等多副本完备删除；2) 确定性删除。源数据及其衍生数据等多副本留存一定时间后必须执行最高程度的删除，删除后源数据及其衍生数据等多副本均不可被恢复。

流通监管包括：1) 契约的执行。数据跨域流通过程涉及数据使用权交易，数据监管方全程监督管理数据提供方与数据接收方的数据提供、交易支付等契约执行情况；2) 问题申诉与受理。数据监管方受理数据提供方与数据接收方因策略契约未执行或执行不到位导致的纠纷与申诉等。

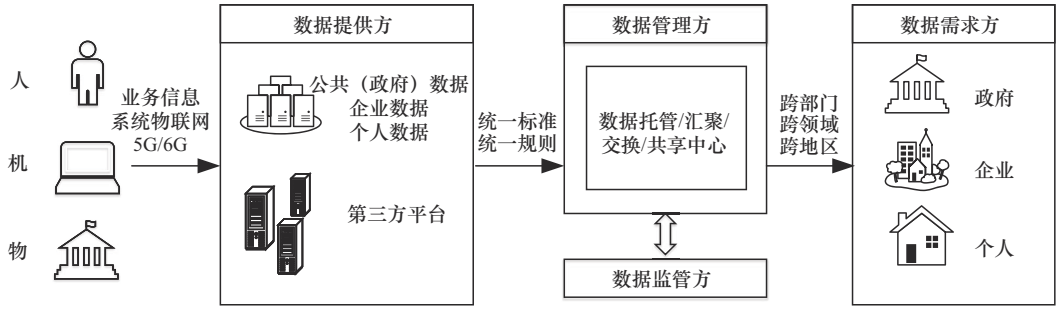


图1　数据流通基本模型

数据流通控制流程如图2所示。首先，数据流通控制中数据使用者（含数据经纪人）需要同数据提供者共同协商数据资源的使用场景（模式）（如开放、共享、交易和交换等）、使用期限、使用成本等，从而形成契约。其次，当契约生效后，数据提供者接受来自数据使用者（含数据经纪人）的数据资源使用请求，将数据资源授权给数据使用者（含数据经纪人）使用。数据使用者（含数据经纪人）则在使用期限内按照既定使用权限（如只读、编辑等）进行加工处理，生成衍生数据。再次，数据使用者（含数据经纪人）接受最终数据使用者的数据资源使用请求，将生成的衍生数据授权流转给最终的数据使用者使用。同样，最终数据使用者在使用期限内按照既定使用权限（如只读、编辑等）进行加工处理，再次生成新的衍生数据。最后，契约到期，数据提供者不再将数据资源授权给数据使用者使用，同时数据使用者（含数据经纪人）也不再将衍生数据授权给最终数据使用者使用，数据使用者（含数据经纪人）删除全部源数据和所有衍生数据关联副本。数据监管者对上述过程全程进行监督管理。

3　基于时间自动机的数据流通控制建模

基于数据资源流通控制的属性和策略，分别对数据流通控制流程、数据提供者实体、数据使用者（含数据经纪人）实体和数据监管者实体等进行形式化方法建模。

3.1　数据流通控制流程建模

数据流通控制模型中位置（状态）、同步通道、变量等符号的含义如表2~表4所示。

表2　数据流通控制模型中位置(状态)及含义

位置(状态)	含义
Start	数据提供者启动数据资源提供
Scene	数据提供者指定具体数据流通场景(模式)
Negotiation	数据提供者与数据使用者协商
Responding	数据提供者响应数据使用者发出的请求
Accept	数据提供者接受数据使用者发出的请求
Send	数据提供者向数据使用者提供数据资源
Idle	空闲状态
Execution	数据使用者按照契约执行
Applying	数据使用者申请数据资源使用
Receiving	数据使用者接收到数据提供者的响应
Owned	数据使用者留存获取的数据资源
DeleteOrigin	数据使用者删除源数据及其衍生数据关联副本
NextResponding	当前数据使用者响应下一个数据使用者发出的请求
NextAccept	当前数据使用者接受下一个数据使用者发出的请求
NextSend	当前数据使用者向下一个数据使用者提供数据资源

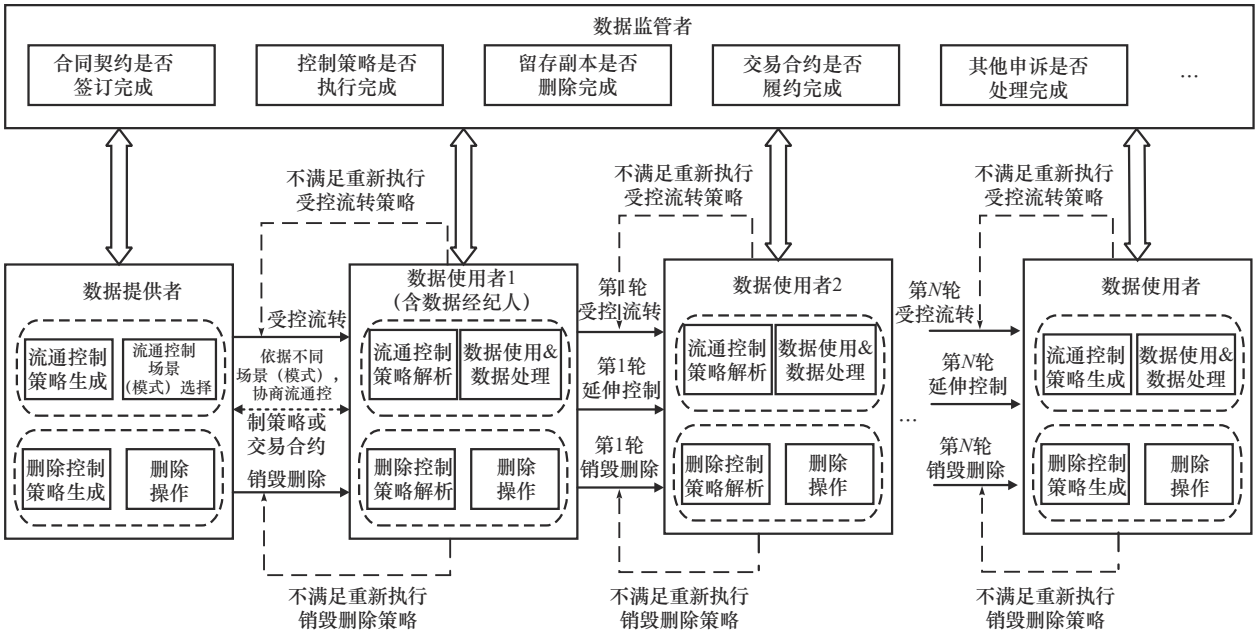


图2　数据流通控制流程

表3 数据流通控制模型中同步通道及含义

同步通道	含义
negotiate	协商数据流通契约内容
negotiate_success	协商成功
negotiate_fail	协商失败
cancel	数据交易取消
trade	数据(分笔)交易成功,如分3笔交易有trade0、trade1和trade2
trade_error	数据交易异常,如监督到2次数据交易异常则有trade1_error和trade2_error
apply	申请提供数据资源
approve	批准数据资源提供申请
reject	驳回数据资源提供申请
send	提供数据资源
free	数据资源提供完成,释放通道
urge	数据交易监督,如对2笔交易进行监督则有urge1和urge2
next_apply	下一个申请提供数据资源
next_approve	批准下一个数据资源提供申请
next_reject	驳回下一个数据资源提供申请
next_send	给下一个申请提供数据资源
next_delete	删除下一个数据使用者产生的衍生数据关联副本
force_delete	强制删除全部源数据和衍生数据关联副本

表4 数据流通控制模型中变量及含义

变量	含义
time_limit	数据资源流通时限
allowed_chain_len	允许数据资源流通传播链长度
farthest_location	数据资源流通可传播的最远位置距离
end_owned	数据使用者是最终交易者
permission	拥有的数据资源访问权限
copies	拥有的衍生数据关联副本数量
copy_id	拥有的衍生数据关联副本ID号
tran_copy_id	下一个数据使用者拥有的衍生数据关联副本ID号
using_times	拥有的衍生数据关联副本使用次数
apply_times	拥有的衍生数据关联副本可申请次数
mode	数据流通场景(模式)
nego_isSuccessful	判断是否协商成功,如数据提供者协商判断变量owner_nego_isSuccessful和数据使用者协商判断变量user_nego_isSuccessful
finish_negotiation	数据提供者与数据使用者协商结束
traded	数据交易完成,如分3笔交易完成有变量traded0、traded1和traded2
trade_cost	数据交易总成本
cost	数据(分笔)交易成本,如分3笔数据交易成本有cost0、cost1和cost2
balance	数据交易账户余额
urge_times	数据交易监督次数,如分2次交易监督次数有urge1_times和urge2_times

数据流通控制流程具体描述如下。

1) 数据提供者与数据使用者（含数据经纪人）共同协商形成数据流通控制策略并达成交易契约，协商内容具体包括数据资源使用可流转轮数、数据资源使用可用剩余次数、数据资源使用期限、数据资源使用场景（模式）、数据资源使用成本、数据资源使用权限等。其中，数据资源使用可流转轮数约定了数据资源流通范围和传播链长度，可流转轮数越多说明数据使用范围越广泛。数据资源使用可用剩余次数约定了数据资源流通使用频率和强度，可用剩余次数越少说明数据使用越频繁。数据资源使用期限约定了数据资源流通时间限制要求。数据资源使用场景（模式）约定了数据资源流通开放、共享、交易和交换4种模式。数据资源使用成本约定了数据资源流通的金额和付款方式。数据资源使用权限约定了数据资源流通主体是否能够打开读取或加工处理数据资源等。

2) 当数据资源使用期限到期后，全部数据使用者（含数据经纪人）均需要销毁删除留存的数据提供者提供的全部数据资源，以及经多轮加工处理后生成的所有衍生数据关联副本。

3) 数据监管者监督管理数据提供者和数据使用者（数据经纪人），具体包括整个数据流通全生命周期契约的签订、执行、履约和投诉过程，并作出相应的处理操作。

3.2 数据提供者实体的建模

数据提供者是数据资源的所有权方。以数据交易场景为例（mode=data_trading），数据提供者Provider需要与数据经纪人Broker共同协商数据交易的具体内容，主要包括交易时限、交易金额、交易方式、交易对象权限、交易传播链长度和衍生数据关联副本数量等。协商成功后按照契约向全部数据使用者提供数据资源，延伸控制交易数据的多次流转，

控制交易源数据及其衍生数据关联副本经多轮交易后依照契约执行销毁删除操作。具体建模过程如下。

1) Provider 调用 owner_negotiate() 和 judge_negotiate() 函数，并通过 negotiate 通道对 time_limit、allowed_chain_len、permission、trade_cost 等变量进行交易协商，交易协商成功后通过 negotiate_success 通道生成契约，通过 trade0 通道完成交易定金支付，交易协商不成功或交易取消后分别通过 negotiate_fail 和 cancel 通道返回 Start 位置（状态），重新进行交易协商，直至交易协商成功。

2) Provider 通过 apply 通道对 Broker 发出的数据资源使用请求进行响应，同意则通过 approve 通道接受 Broker 发出的数据资源使用请求，Provider 从 Responding 位置（状态）转为 Accept 位置（状态），拒绝申请则通过 reject 通道返回 Start 位置（状态）。

3) 在数据资源使用期限内，Provider 通过 send 通道向 Broker 提供数据资源，通过 trade1 通道完成交易首付款支付，并转为 Send 位置（状态），当时限到期后，调用 reset() 函数，通过 free 通道释放数据交易进程并回到 Start 位置（状态）。

建立 Provider 时间自动机 $< L,l_0,C,\Sigma,E,I >$，其中，位置（状态）集合即 L={Start,Scene,Negotiation, Responding, Accept, Send, Urgent, …}，初始位置（状态）集合 l_0={Start}。Provider 时间自动机模型如图 3 所示。

数据交易协商策略包括每一轮交易每个数据使用者（含数据经纪人）的访问控制权限（如只读打开、增加编辑、复制副本等），每一轮交易每个数据使用者（含数据经纪人）产生衍生数据关联副本的数量、数据交易场景（模式）下约定定金、首付款和尾款比例等，Provider 与 Broker 进行交易契约协商如算法 1 所示。

算法 1　Provider 与 Broker 进行交易契约协商

```
begin
    //初始化参数
    time_limit←32 000
    allowed_chain_len←2
    //设置第 1 轮 Broker 的数据访问权限
    permissions[0][OPEN]←1 //允许只读
    permissions[0][ADD_EDIT]←1//允许添加/编辑
    permissions[0][COPY]←1 //允许复制
    //设置第 2 轮 Consumer 的数据访问权限
    permissions[1][OPEN]←1 //允许只读
    permissions[1][ADD_EDIT]←1//允许添加/编辑
    permissions[1][COPY]←1 //允许复制
    //设置每轮的可复制次数
    copies[0]←3 //第 1 轮：Broker
    copies[1]←3 //第 2 轮：Consumer
    //设置数据交易模式
    mode←3
    //如果是数据交易模式，则设定各类交易成本
    if mode=3 then
      trade_cost←100 //设定交易总成本
      cost0←20 //设定第 1 笔交易成本
      cost1←50 //设定第 2 笔交易成本
```

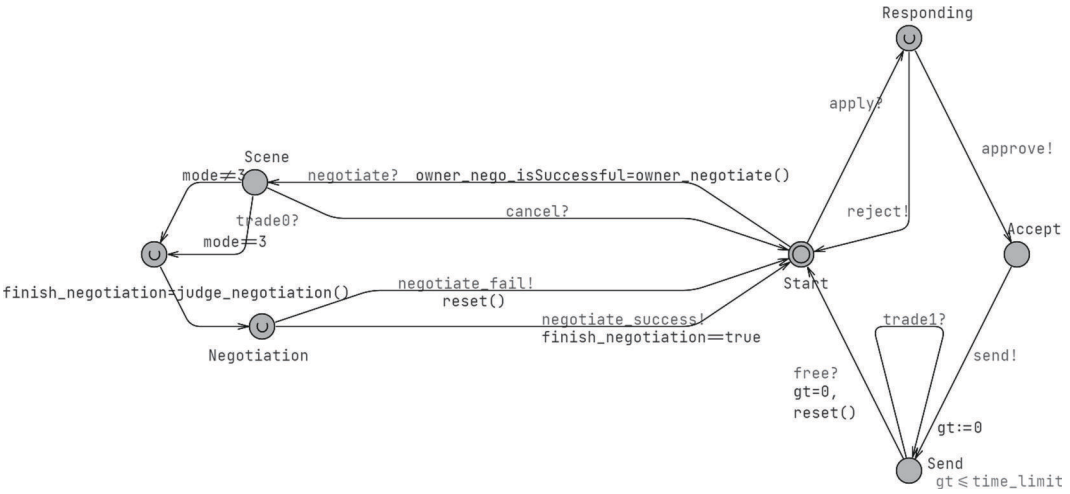


图 3　Provider 时间自动机模型

cost2←30 //设定第3笔交易成本
end if
return true //返回交易契约协商成功状态
end

3.3 数据使用者(含数据经纪人)实体的建模

数据使用者（含数据经纪人）是数据资源的使用权方。以上述数据交易场景为例，一方面，数据经纪人Broker接收到数据提供者Provider提供的数据资源，按照契约留存数据资源加工处理，产生衍生数据关联副本，并向下一个数据使用者Consumer提供数据资源。另一方面，数据交易时限到期后，数据经纪人Broker删除留存的源数据Origin和加工处理源数据产生的衍生数据关联副本OriginCopies。同样，下一个数据使用者Consumer也删除留存的源数据Origin和衍生数据关联副本OriginCopies。数据监管者Supervisor对数据使用者和数据交易支付全程进行监督管理。具体建模过程如下。

1) Broker调用user_negotiate()函数，通过negotiate、negotiate_sucess、cancel等通道配合完成与Provider的交易协商或取消，当交易账户余额大于定金时（即balance≥cost0），调用pay0()函数通过trade0通道完成交易定金支付，协商不成功则调用refund()函数通过negotiate_fail通道退还定金。

2) Broker通过apply、approve、reject等通道配合完成与数据提供者Provider的申请确认。

3) Broker通过send通道接收Provider提供的数据资源，从Receiving位置（状态）转到Owned位置（状态）。同时，当交易账户余额大于首付款时（即balance≥cost1），调用pay1()函数通过trade1通道完成交易首付款支付，如果始终未支付交易尾款则通过urge通道接受Supervisor督促。

4) Broker根据数据交易契约，按照约定好的数据使用权限调用open()函数进行只读操作，调用edit()函数进行编辑操作，产生相应的衍生数据关联副本。同时，调用copies_add()函数对产生的衍生数据关联副本的数量进行记录。

5) Broker根据数据交易契约，① 如果数据交易使用传播链长度allowed_chain_len=1，则当前Broker即最终数据使用者。② 如果数据交易使用传播链长度allowed_chain_len=2，则Broker以新的数据提供者身份延伸控制交易数据资源二次流转，通过next_apply、next_approve、next_reject等通道配合完成与Consumer的申请确认，通过next_send通道完成与Consumer的数据提供，Broker先从NextResponding位置（状态）转为NextAccept位置（状态），最后从NextSend位置（状态）回到Owned位置(状态)，同时Consumer通过next_apply、next_approve、next_reject、next_send等通道配合完成Broker数据接收，Consumer根据数据交易契约，按照约定好的数据使用权限分别调用open()函数和edit()函数进行相应的只读和编辑操作，同时产生相应的OriginCopies，当前Consumer即最终数据使用者。③ 如果数据交易使用传播链长度allowed_chain_len＞2，则与②类似，Broker以新的数据提供者身份通过next_apply、next_approve、next_reject、next_send等通道配合完成与下一个数据使用者Consumer*X*的数据提供，Broker完成从NextResponding位置（状态）到NextAccept位置（状态）、NextSend位置（状态）回到Owned位置（状态）的转移，Consumer*X*与下一个数据使用者Consumer*Y*重复②的步骤进行多轮交易，延伸控制交易数据多次流转，直至遇到最终数据使用者Consumer。

6) 根据数据交易契约，提供的数据资源使用时限到期，① 如果数据交易使用传播链长度allowed_chain_len=1，则当前Broker即最终数据使用者，删除Origin并调用copy_del()函数删除自身产生的OriginCopies，调用pay2()函数通过trade2通道完成交易尾款支付，Broker从Owned位置（状态）转为DeleteOrigin位置（状态），最后通过free通道释放数据交易进程并回到Idle位置（状态），按照契约完成销毁删除操作，履约完成则交易终结。② 如果数据交易使用传播链长度allowed_chain_len=2，Broker删除Origin并调用copy_del()函数删除自身产生的OriginCopies，调用tran_copy_del()函数通过next_delete通道删除因二次流转由Consumer产生的OriginCopies，调用pay2()函数通过trade2通道完成交易尾款支付，Broker从Owned位置（状态）转为DeleteOrigin位置（状态），最后通过free通道释放数据交易进程并回到Idle位置（状态），当前Consumer即最终数据使用者，按照契约完成销毁删除操作，履约完成则交易终结。③ 如果数据交易使用传播链长度allowed_chain_len＞2，则与②类似，Broker删除Origin并调用copy_del()、tran_copy_del()等函数通过next_delete通道删除产生的全部Origin-

Copies，Consumer$X$与下一个数据使用者Consumer$Y$重复②的步骤，直至遇到最终数据使用者Consumer，调用pay2()函数通过trade2通道完成交易尾款支付，Broker从Owned位置（状态）转为DeleteOrigin位置（状态），最后通过free通道释放数据交易进程并回到Idle位置（状态），按照契约完成销毁删除操作，履约完成则交易终结。

值得说明的是，针对Broker交易首付款和尾款支付通过urge通道接受Supervisor监督管理：首付款支付3次督促提醒后仍未支付则删除Origin并通过trade1_error通道处理，Broker从Owned位置（状态）转为DeleteOrigin位置（状态），最后通过free通道释放数据交易进程并回到Idle位置（状态）；尾款支付3次督促提醒后仍未支付则删除Origin并通过trade2_error通道、forced_del通道和next_delete通道配合强制删除全部衍生数据关联副本OriginCopies，最后通过free通道释放数据交易进程并回到Idle位置（状态），契约交易支付策略未履约完成且终止执行。

分别建立Broker时间自动机$< L,l_0,C,\Sigma,E,I >$和Consumer时间自动机$< L,l_0,C,\Sigma,E,I >$。其中，Broker时间自动机的位置（状态）集合L={Idle,Negotiation, Execution, Applying, Receiving, Owned, Urgent, Committed,…}，初始位置（状态）集合l_0={Idle}，Broker时间自动机模型如图4所示。

Consumer时间自动机的位置（状态）集合L={Idle, Applying, Receiving, Owned, Urgent, Committed, …}，初始位置（状态）集合l_0={Idle}，Consumer时间自动机模型如图5所示。

数据使用者（含数据经纪人）产生衍生数据关联副本如算法2所示。

算法2　数据使用者（含数据经纪人）产生衍生数据关联副本

```
begin
    //增加副本，并存入副本数列表
    list[len][0]←copy_id+1
    copy_id←copy_id+1
    //检查数据访问权限
    if permissions[1][1]=1 and permissions[1]
      [2]=1 then
       list[len][1]←EDIT
    else
       list[len][1]←OPEN
    end if
    //更新副本数列表的长度
    len←len+1
    //记录最新副本数量
    tran_copy_id←copy_id
end
```

数据使用者（含数据经纪人）删除衍生数据关联副本如算法3所示。

算法3　数据使用者（含数据经纪人）删除衍生数据关联副本

```
begin
```

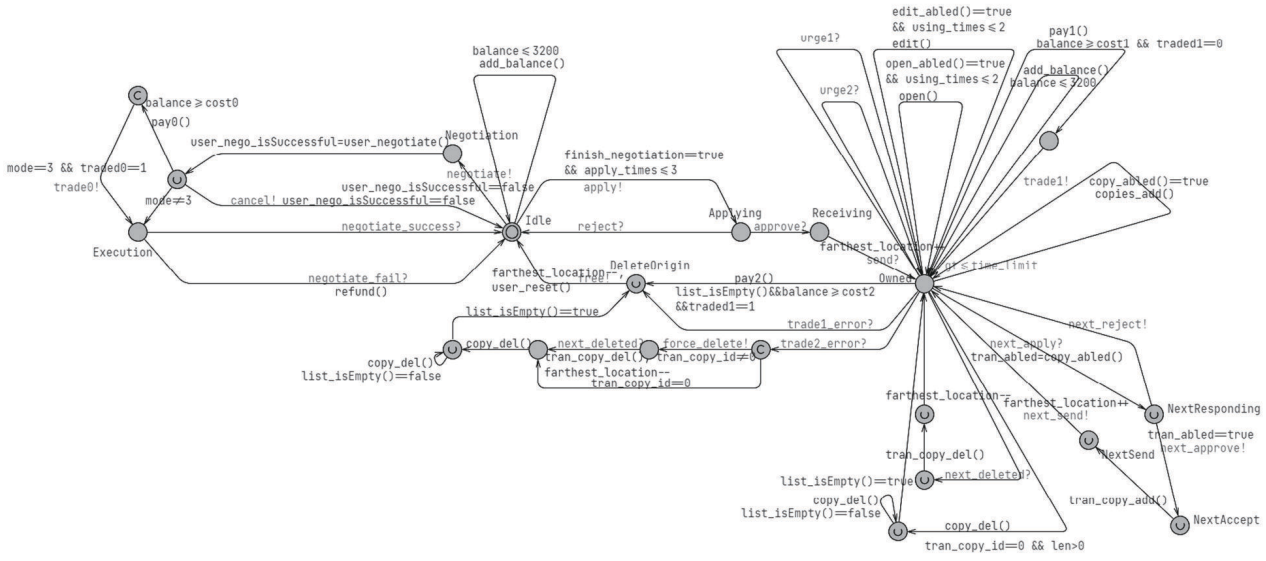


图4　Broker时间自动机模型

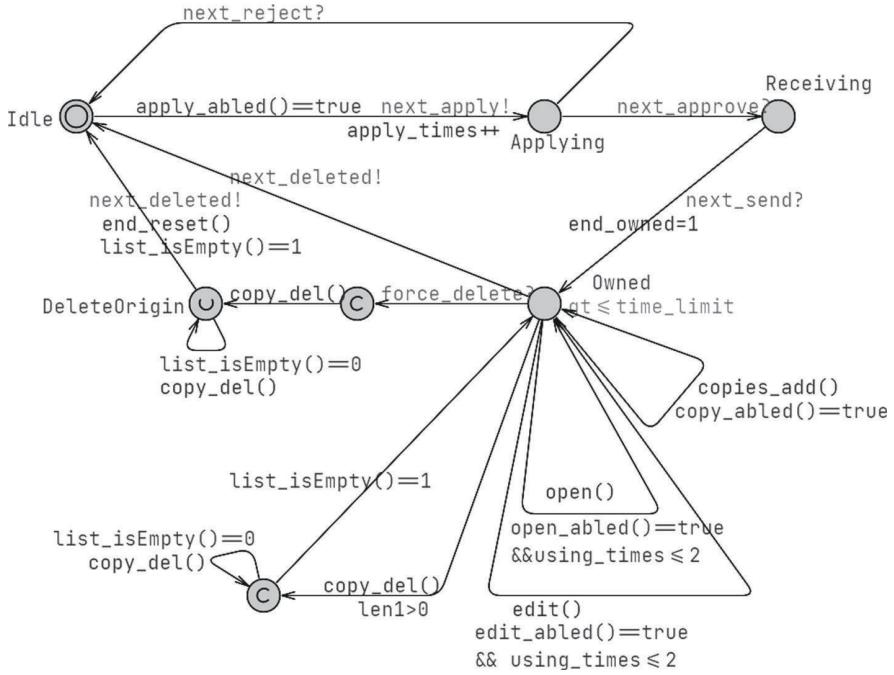


图5　Consumer时间自动机模型

```
//记录当前的副本数
t←tran_copy_id
i←0
//在副本数列表中查找OriginCopies
while i < len and list[i][0]≠t do
   i←i+1
end while
//如果找到OriginCopies，则删除且副本数列表向前移动
while i < len−1 do
   list[i]←list[i+1]
   i←i+1
end while
//直至删除最后一个OriginCopies
list[i][0]←0
len←len − 1
//重置副本数
tran_copy_id←0
end
```

3.4　数据监管者实体的建模

数据监管者是数据资源的监管权方。以上述数据交易场景为例，数据监管者Supervisor全程监督管理数据交易支付行为。具体建模过程如下。

1) Supervisor针对Broker交易首付款和尾款支付，通过urge1通道和urge2通道分别完成对Broker的支付首付款督促和支付尾款督促，默认每笔款支付最多督促提醒3次。

2) Supervisor针对Broker交易首付款和尾款始终未支付，通过trade1_error通道完成Origin删除，通过trade2_error通道完成Origin及OriginCopies删除，Supervisor回到Idle位置（状态）。

建立Supervisor时间自动机 $< L,l_0,C,\Sigma,E,I >$，其中，位置（状态）集合 $L=\{\text{Idle,Committed,}\cdots\}$，初始位置（状态）集合 $l_0=\{\text{Idle}\}$，Supervisor时间自动机模型如图6所示。

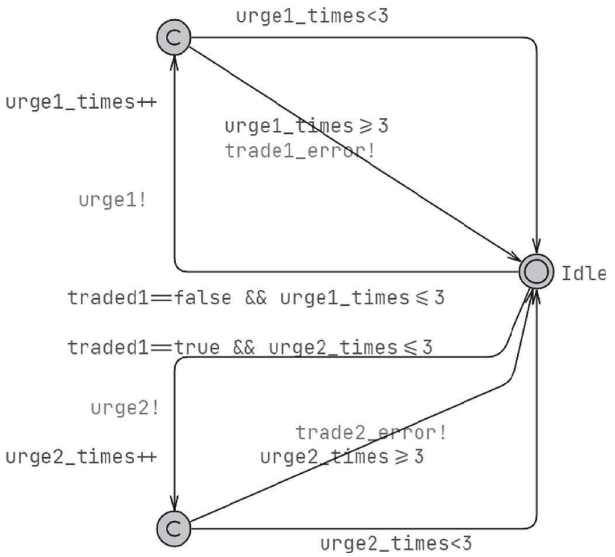


图6　Supervisor时间自动机模型

4　基于时序逻辑形式化描述与分析

本文将数据流通控制的计算树逻辑与线性时序逻辑分解为安全需求性质和流通控制属性两大类，统一采用CTL*公式进行形式化描述与验证。

4.1　安全需求性质

1) 无死锁。确保在数据资源流通过程中，数据提供者、数据使用者（含数据经纪人）和数据监管者在任何情况下都不会进入死锁状态。

验证语句：*A*[] not deadlock

若验证结果通过，则满足该性质。

2) 完整性。即可终止，确保数据资源流通过程中所有执行能够正确地终止，最终回到正确状态。例如：数据交易结束，数据提供者、数据使用者（含数据经纪人）和数据监管者均回到初始位置（状态）。

验证语句：*A*<> (Provider.Start && Broker.Idle && Consumer.Idle && Supervisor.Idle)

若验证结果通过，则满足该性质。

3) 公平性。即确保执行过程中，必然会到达它可以到达的位置。例如：数据使用者（含数据经纪人）申请数据资源，数据提供者一旦批准，数据使用者（含数据经纪人）则一定能收到申请的数据资源。

验证语句：*A*<> (Provider.Accept imply Broker.Owned)

若验证结果通过，则满足该性质。

4) 活动性。即确保执行某个操作后，不会一直停留在当前位置（状态）。例如：数据使用者（含数据经纪人）只能在授权时限内使用数据资源，而不能拥有无限时的数据资源。

验证语句：*A*<> not Broker.Applying

若验证结果通过，则满足该性质。

5) 一致性。即确保在所有路径上的所有位置（状态）中，如果系统处于某个状态，那么变量一定赋值成功。例如：数据交易开始后，数据提供者始终处于初始位置（状态），说明数据交易未执行，数据资源最远传播位置变量赋值为0。

验证语句：*A*[] (Provider.Start imply farthest_location==0)

若验证结果通过，则满足该性质。

4.2　流通控制属性

数据流通控制对数据资源使用期限内远程延伸控制、到期销毁删除和履行交易契约等有着明确约束。因此，在数据交易场景（模式）下，分别从流通控制流程、数据提供者、数据使用者（含数据经纪人）和数据监管者等视角，对流通控制属性的计算树时序逻辑进行描述、验证和分析，并给出反例验证。

1) 延伸控制性。确保数据资源在整个流通时限内，只能按照契约协商确定的策略进行数据交易，多轮交易后数据依照策略契约使用。例如：协商约定“交易传播链长度为1，且最终数据使用者权限为只读（不允许进行编辑和产生衍生数据关联副本）”，数据交易只能按照该策略进行，验证语句如下。

A[] Consumer.apply_abled()==false

A[] Consumer.edit_abled()==false

A[] Consumer.copy_abled()==false

若验证结果通过，则满足该性质。

2) 交易时限性。确保数据资源在整个流通时限内完成全部交易。例如：在数据交易时限内，数据提供者始终能够向数据使用者（含数据经纪人）提供数据资源，验证语句如下。

A[] (Provider.Send imply gt<=time_limit)

A<> (Provider.Start && gt<=time_limit)

若验证结果通过，则满足该性质。

此处，给出反例验证语句如下。

E<> (Consumer.DeleteOrigin && gt>time_limit)

若验证结果不通过，则上述反例不满足该性质，说明不存在超出数据流通时限的数据交易。

3) 删除确定性。确保数据资源流通时限到期后，数据使用者（含数据经纪人）留存的全部源数据和衍生数据关联副本都被销毁删除。例如：数据交易时限到期后，数据使用者（含数据经纪人）留存的全部源数据和衍生数据关联副本均已被销毁删除，验证语句如下。

A<> (Provider. Start && Broker. list_isEmpty() && Consumer.list_isEmpty())

Broker.Owned-->(Broker.DeleteOrigin && gt<=time_limit)

若验证结果通过，则满足该性质。

此处，给出反例验证语句如下。

E<> (Broker.DeleteOrigin && gt>time_limit)

若验证结果不通过，则上述反例不满足该性质，说明不存在数据交易时限到期，数据使用者

（含数据经纪人）留存的全部源数据及其衍生数据关联副本仍未被销毁删除的情况。

4) 履约完整性。确保在数据资源流通过程中，数据使用者（含数据经纪人）能够按照交易契约履行付款义务。例如：对数据使用者（含数据经纪人）而言，数据交易结束后必须及时删除全部数据，否则无法支付尾款，验证语句如下。

E<> (Broker.list_isEmpty()==0 && traded2==1)

若验证结果通过，则满足该性质。

此处，给出反例1验证语句如下。

E<> (traded1==0 && Broker.open_abled()==1)

E<> (traded1==0 && Broker.edit_abled()==1)

E<> (traded1==0 && Broker.copy_abled()==1)

若验证结果不通过，则上述反例不满足该性质，说明不存在数据使用者（含数据经纪人）未支付首付款就可以使用数据资源（包括只读、编辑、产生衍生数据关联副本）的情况。

此处，再次给出反例2验证语句如下。

E<> (traded1==0 && traded1==1)

若验证结果不通过，则上述反例不满足该性质，说明不存在未支付首付款却可以支付尾款的情况。

5) 监管时效性。确保在数据资源流通过程中，数据监管者能够及时监督和处理数据交易失败的情况。例如，数据监管者对交易首付款和尾款支付分别进行督促，如果超过其督促提醒次数上限，则强制数据使用者（含数据经纪人）删除留存的源数据及其衍生数据关联副本，验证语句如下。

A<> Supervisor. urge1_times>=3 imply Broker. DeleteOrigin

A<> Supervisor. urge1_times>=3 imply Consumer.DeleteOrigin

A<> Supervisor. urge2_times>=3 imply Broker. DeleteOrigin

若验证结果通过，则满足该性质。

5 仿真与验证

5.1 关于UPPAAL

UPPAAL[30]是一个集形式化方法建模、验证和分析实时系统的高效验证工具，该工具基于时间自动机模型且支持CTL和LTL，通过模型检查和验证系统的时序行为是否符合要求，计算树时序逻辑公式及含义如表5所示。

表5 计算树时序逻辑公式及含义

公式	含义
E<>*p*	存在一条路径，最终成立*p*
A[]*p*	对于所有路径，始终成立*p*
E[]*p*	存在一条路径，始终成立*p*
A<>*p*	对于所有路径，始终成立*p*
p→*q*	每当成立*p*时，最终将成立*q*

用户通过使用编辑器自定义状态、时钟、变量、转换规则等，通过图形界面创建时间自动机模型，在验证器中输入CTL和LTL公式进行检查验证，在模拟器中查看系统在不同时间点的时序行为，跟踪时钟和变量的变化，从而完成实时系统的检查验证、调度优化、资源分配等动作。

5.2 仿真验证结果

根据上述建模和形式化描述，将由数据提供者、数据使用者（含数据经纪人）和数据监管者实体所构成的数据流通控制时间自动机网络模型输入UPPAAL验证工具编辑器，在其模拟器和验证器中分别进行调试、仿真与验证。

如图7所示，UPPAAL给出了仿真运行轨迹和位置（状态）转换过程。其中，左侧模拟Trace偶数行表示数据交易动作，奇数行记录各时间自动机位置（状态）的变化，右侧表示位置（状态）转换过程。

仿真完成后，对上述建立的基于时间自动机的数据流通控制模型通过UPPAAL验证器进行安全需求性质和流通控制属性验证，验证结果如表6所示。

6 结束语

为了解决数据跨域流通控制机制验证难题，本文提出了基于时间自动机和计算树时序逻辑的形式化建模及验证方法，并利用UPPAAL验证工具进行了仿真验证。结果分析表明，该方法可以有效验证数据流通控制策略生成、传递与执行的可行性、正确性和安全性。本文方法的局限性在于以下两方面。一方面，当使用UPPAAL验证工具对时间自动机网络进行形式化方法验证时，随着实体模型自动机数量、位置（状态）数量和时钟变量的增多，验证运行轨迹和转换过程复杂程度增高，运算资源和耗时大大增加，甚至可能因算力耗尽导致验证失败。因此，在形式化建模过程中对部分变量和函数进行了

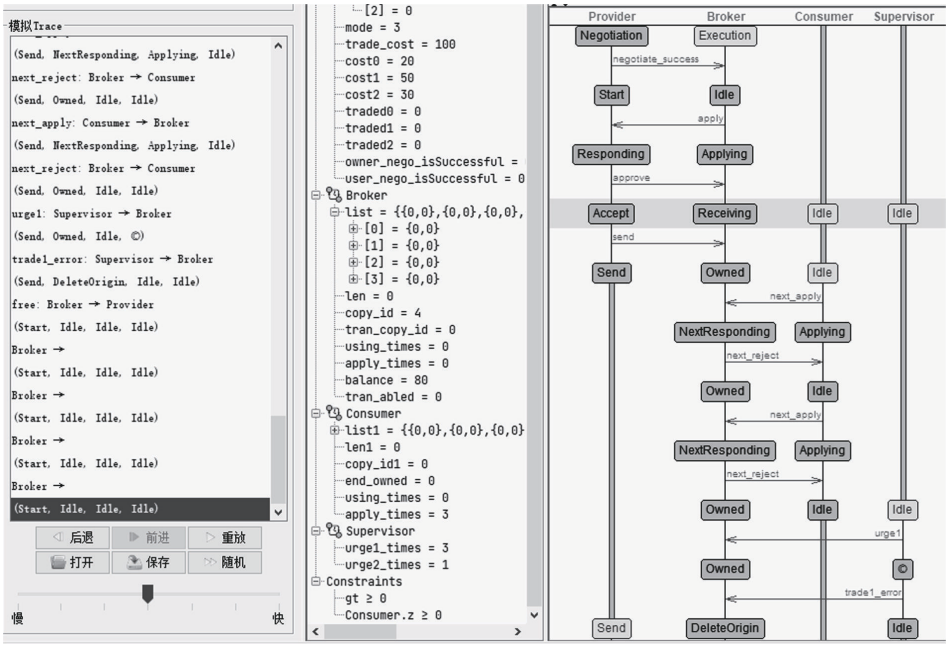


图7　数据交易场景(模式)下数据流通控制流程UPPAAL模拟器仿真结果

表6　　数据交易场景(模式)下数据流通控制可行性、正确性和安全性验证结果

验证性质	CTL表达式	验证时间/s	验证结果
无死锁	*A*[] not deadlock	214.234	满足
完整性	*A*<> (Provider.Start && Broker.Idle && Consumer.Idle && Supervisor.Idle)	0.006	满足
公平性	*A*<> (Provider.Accept imply Broker.Owned)	0.016	满足
活动性	*A*<> not Broker.Applying	0.006	满足
一致性	*A*[] (Provider.Start imply farthest_location==0)	44.781	满足
延伸控制性	*A*[] Consumer.apply_abled()==false	117.274	满足
延伸控制性	*A*[] Consumer.edit_abled()==false	119.260	满足
延伸控制性	*A*[] Consumer.copy_abled()==false	122.378	满足
交易时限性	*A*[] (Provider.Send imply gt<=time_limit)	39.718	满足
交易时限性	*A*<> (Provider.Start && gt<=time_limit)	56.172	满足
交易时限性(反例)	*E*<> (Consumer.DeleteOrigin && gt>time_limit)	341.453	不满足
删除确定性	*A*<> (Provider.Start && Broker.list_isEmpty() && Consumer.list_isEmpty())	116.609	满足
删除确定性	Broker.Owned→(Broker.DeleteOrigin && gt<=time_limit)	233.571	满足
删除确定性(反例)	*E*<> (Broker.DeleteOrigin && gt>time_limit)	275.390	不满足
履约完整性	*E*<> (Broker.list_isEmpty()==0 && traded2==1)	311.418	满足
履约完整性(反例)	*E*<> (traded1==0 && Broker.open_abled()==1)	221.189	不满足
履约完整性(反例)	*E*<> (traded1==0 && Broker.edit_abled()==1)	222.340	不满足
履约完整性(反例)	*E*<> (traded1==0 && Broker.copy_abled()==1)	220.198	不满足
履约完整性(反例)	*E*<> (traded1==0 && traded1==1)	372.765	不满足
监管时效性	*A*<> Supervisor.urge1_times>=3 imply Broker.DeleteOrigin	0.003	满足
监管时效性	*A*<> Supervisor.urge1_times>=3 imply Consumer.DeleteOrigin	0.002	满足
监管时效性	*A*<> Supervisor.urge2_times>=3 imply Broker.DeleteOrigin	0.078	满足

约束和精简，提高验证效率的同时对数据流通控制真实还原度有一定影响。另一方面，受限于UPPAAL验证工具缺少变量传递与外部接口，形式化方法建模和验证过程中未考虑数据流通控制策略可信执行问题。未来将尝试借助其他验证工具对在可信环境中进行细粒度日志审计和合规性检测进行形式化方法建模与验证。

参考文献：

[1] 李凤华, 李晖, 牛犇, 等. 数据要素流通与安全的研究范畴与未来发展趋势[J]. 通信学报, 2024, 45(5): 1-11.
LI F H, LI H, NIU B, et al. Research category and future development trend of data elements circulation and security[J]. Journal on Communications, 2024, 45(5): 1-11.

[2] ALUR R, DILL D L. A theory of timed automata[J]. Theoretical Computer Science, 1994, 126(2): 183-235.

[3] LARSEN K G, PETTERSSON P, YI W, et al. UPPAAL: a tool for automatic verification of real-time systems[C]//Proceedings of the 10th International Conference on Computer-Aided Verification (CAV). Berlin: Springer, 1997: 456-460.

[4] RAJSKI K, JANSON S. Efficient verification of timed automata using UPPAAL[C]//Proceedings of the 13th International Conference on Computer-Aided Verification (CAV). Berlin: Springer, 2001: 103-114.

[5] CHECHIK M, KATZ M. Timed automata and timed arc petri nets: verification and application[C]//Proceedings of the 16th International Conference on Computer-Aided Verification (CAV). Berlin: Springer, 2006: 325-338.

[6] KATOEN J P, KWIATKOWSKA M. Verification of timed automata using model checking[C]//Proceedings of the 14th International Conference on Computer-Aided Verification (CAV). Berlin: Springer, 2003: 1-14.

[7] BEHRMANN G, DAVID A, LARSEN K G, et al. UPPAAL 4.0[C]//Proceedings of the 3rd International Conference on the Quantitative Evaluation of Systems. New York: ACM Press, 2006: 125-126.

[8] BENGTSSON J, YI W. Timed automata: semantics, algorithms and tools[C]//Advanced Course on Petri Nets. Berlin: Springer, 2004: 87-124.

[9] LIN H M, YI W. A proof system for timed automata[C]//International Conference on Foundations of Software Science and Computation Structures. Berlin: Springer, 2000: 208-222.

[10] LIN H M, YI W. Axiomatising timed automata[J]. Acta Informatica, 2002, 38(4): 277-305.

[11] ZHAO J H, LI X D, ZHENG T, et al. Removing irrelevant atomic formulas for checking timed automata efficiently[C]//International Conference on Formal Modeling and Analysis of Timed Systems. Berlin: Springer, 2004: 34-45.

[12] ZHAO J H, LI X D, ZHENG G L. A quadratic-time DBM-based successor algorithm for checking timed automata[J]. Information Processing Letters, 2005, 96(3): 101-105.

[13] CHECHIK M, KATZ M. Efficient timed automata verification for cyber-physical systems[C]//Proceedings of the 33rd International Conference on Computer-Aided Verification (CAV). Berlin: Springer, 2021: 109-122.

[14] BJØRNER D, JANSEN A. Timed automata and quantitative model checking for embedded systems[C]//Proceedings of the 24th International Conference on Formal Engineering Methods (ICFEM). Berlin: Springer, 2022: 116-130.

[15] LARSEN K G, RASKIN J F. Timed automata with probabilistic extensions: model checking and applications[C]//Proceedings of the 28th International Conference on Automated Planning and Scheduling (ICAPS). Berlin: Springer, 2023: 347-359.

[16] PÉREZ M, GARCÍA A. Using UPPAAL for autonomous vehicle systems verification[J]. IEEE Transactions on Vehicular Technology, 2009, 72(2): 122-135.

[17] WANG X, LI Y. Symbolic timed automata: efficient verification and applications in CPS[C]//Proceedings of the 23rd International Conference on Cyber-Physical Systems (ICCPS). Piscataway: IEEE Press, 2022: 101-114.

[18] SUN H, YANG M. Integration of timed automata and hybrid systems for real-time embedded system verification[J]. Formal Methods in System Design, 2023, 62(3), 203-218.

[19] ZHAO H, ZHANG L. Timed automata and game theory: applications to multi-agent systems[C]//Proceedings of the 10th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). Piscataway: IEEE Press, 2022: 1559-1570.

[20] ZHAO W, ZHANG Y. UPPAAL and symbolic model checking: advancements and applications in cyber-physical systems[C]//Proceedings of the 25th International Conference on Computer-Aided Verification (CAV). Berlin: Springer, 2023: 68-82.

[21] LIU Z, CHEN Y. UPPAAL for IoT system security and verification[J]. IEEE Access, 2022: 10, 5123-5137.

[22] CLARKE E M, EMERSON E A, SISTLA A P. Automatic verification of finite state concurrent system using temporal logic specifications[C]//Proceedings of the 10th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages. New York: ACM Press, 1983: 117-126.

[23] PNUELI A. The temporal logic of programs[C]//Proceedings of the 18th Annual Symposium on Foundations of Computer Science. Piscataway: IEEE Press, 1977: 46-57.

[24] EMERSON E A, HALPERN J Y. “Sometimes” and “not never” revisited: on branching versus linear time temporal logic[J]. Journal of the ACM (JACM), 1986, 33(1): 151-178.

[25] BALDI G, DIAZ-TELLEZ Y, DIMITRAKOS T, et al. Session-dependent usage control for big data[J]. Journal of Internet Services and Information Security, 2020, 10(3): 76-92.

[26] CHEUNG H, YANG C G, CHEUNG H. New smart-grid operation-based network access control[C]//Proceedings of the 2015 IEEE Energy Conversion Congress and Exposition (ECCE). Piscataway: IEEE Press, 2015: 1203-1207.

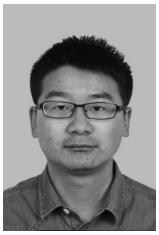
[27] BERNABE J B, RAMOS J L H, GOMEZ A F S. TACIoT: multidimensional trust-aware access control system for the Internet of things[J]. Soft Computing, 2016, 20(5): 1763-1779.

[28] QI H, MA H X, LI J Q, et al. Access control model based on role and attribute and its applications on space-ground integration networks[C]// Proceedings of the 2015 4th International Conference on Computer Science and Network Technology (ICCSNT). Piscataway: IEEE Press, 2015: 1118-1122.

[29] PARK J, SANDHU R. The $UCON_{ABC}$ usage control model[J]. ACM Transactions on Information and System Security, 2004, 7(1): 128-174.

[30] LARSEN K G, RASKIN J F. Timed automata: semantics and verification[C]//Proceedings of the 13th International Conference on Computer Aided Verification (CAV). Berlin: Springer, 2001: 1-22.

[作者简介]



李恒（1990– ），男，陕西汉中人，中国科学院信息工程研究所博士生、高级工程师，主要研究方向为数据要素流通、访问控制、信息保护、隐私计算等。

李凤华（1966– ），男，湖北浠水人，博士，中国科学院信息工程研究所研究员、博士生导师，主要研究方向为网络与系统安全、信息保护、隐私计算等。

梁琬珩（2003– ），女，广东湛江人，中国科学院信息工程研究所博士生，主要研究方向为访问控制。

郭云川（1977– ），男，四川营山人，博士，中国科学院信息工程研究所高级工程师、博士生导师，主要研究方向为访问控制。

张玲翠（1986– ），女，河北故城人，博士，中国科学院信息工程研究所高级工程师，主要研究方向为网络与系统安全。

周紫妍（1998– ），女，河北秦皇岛人，中国科学院信息工程研究所博士生，主要研究方向为访问控制。